

# **Oracle Banking Platform**

Extensibility Guide – Adapter Usage Details

Release 2.3.0.0.0

**E56276-01**

July 2014

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

**Preface .....iv**

- Audience .....iv
- Documentation Accessibility .....iv
- Related Documents .....iv
- Conventions .....iv

**1 Adapter Usage Details .....1**

---

---

# Preface

Oracle Banking Platform Extensibility - Adapter Usage Details guide is an extension of the Oracle Banking Platform Extensibility guide. It contains information about the detailed list of adapters which can be used for extending and customizing the product.

This preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

This guide is intended for the users of Oracle Banking Platform.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see the following documentation:

- For information related to customization and extension of Oracle Banking refer the Oracle Banking Extensibility Guide

## Conventions

The following text conventions are used in this document:

---

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for

---



---

which you supply particular values.

monospace

Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---



## Adapter Usage Details

This chapter provides information on the list of adapters which can be used for extending and customizing the product.

**Table 1-1 Adapter Usage Details**

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table <b>FLX_DA_ACCT_EVENT_B</b> which will have one to one relationship with behaviour<b>FLX_DA_ACCT_BEHAVIOUR_B</b> and one to many relationship with details <b>FLX_DA_ACCT_EVENT_DTLS_B</b>.   @param accountingRequest {@link ClassificationAccountingEvent} object containing details of accounting event	This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table <b>FLX_DA_ACCT_EVENT_B</b> which will have one to one relationship with behaviour<b>FLX_DA_ACCT_BEHAVIOUR_B</b> and one to many relationship with details <b>FLX_DA_ACCT_EVENT_DTLS_B</b>.   @param accountingRequest {@link ClassificationAccountingEvent} Object containing details of accounting event for asset classification @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.ac.dto.accounting.AssetClassificationAccountingEventDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param accountingRequest {@link ClassificationAccountingEvent} object containing details of accounting event</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param accountingRequest {@link ClassificationAccountingEvent} object containing details of accounting event</p>	<p>@param sessionContext @param applicationContext @param loanAccount @param amountTagDetailsDTOs @param eventSequenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param accountingRequest {@link ClassificationAccountingEvent} object containing details of accounting event</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	@param accountingEventDTO	This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table <b>FLX_DA_ACCT_EVENT_B</b> which will have one to one relationship with behaviour<b>FLX_DA_ACCT_BEHAVIOUR_B</b> and one to many relationship with details <b>FLX_DA_ACCT_EVENT_DTLS_B</b>. @param accountingRequest {@link ClassificationAccountingEvent} Object containing details of accounting event for asset classification @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.ac.dto.accounting.AssetClassificationAccountingEventDTO)
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	@param accountingEventDTO	
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	@param accountingEventDTO	@param sessionContext @param applicationContext @param loanAccount @param amountTagDetailsDTOs @param eventSequenceNumber
IAssetClassificationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountingAdapter	raiseAccountingEvent	@param accountingEventDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This method is called to save the a Draft Record of PartyMerge Request.  This is required for Audit purpose also during the Transaction the Party Merged state can be retrieved using this draft record. <ul> <li>Validation is performed on Input. If PartyMergeDTO or any of the merging partyIds is null, Exception is thrown. <li>The method can be called from two perspectives. <ul> <li>The Draft already exists but it needs to get updated (a valid mergeId is given). <ul> <li>PartyMergeState entity is directly loaded from the Party Merge state table <li>If no such record exists in the DB against given mergeId, a partyMergeState is created which and persisted into the DB. If the record Exists, it gets updated with the new data and values which comes as input to the method. </ul> <li>A new Draft needs to be created.(No mergeId given) <ul> <li>Victim party status id set to <i>MERGE_INITIATED</i> which gets persisted. <li>Then required Party details record for survivor and victim for merge is collected. <li>With these data Party Merge State entity is created and persisted into DB. </ul> </ul>
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This method will create the workflow tracker summarized entity. @param mergeId @param partyMergeDTO @param partyMergeStatus



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method is used for initiating merge. The Merge process follows sequential steps as below: <ul><li>First validation of Input Parameter (PartyMergeDTO) is done. <li>If PartyDTO (IndividualDTO for Party type Individual, TrustDTO for Trust, OrganizationDTO for Legal Entities) in the PartyMergeDTO is null, then it by default gets set to corresponding value of survivor's. <li>In case mergeId is null, that means SaveAsDraft has not been called before for this particular party merge, so saveAsDraft is called. <li>After all the data are available for the merge to proceed, Validation of Party Merge Business Policy is done. Any error occurred during this validation gets recorded in partyMergeResponse. <li>Now, based on the purpose for which the method has been called, the behavior is as below: <li>Initiate Party Merge (Actual Party Merge) <ul><li>Validation is performed for party data, i.e. comparison is done for data that comes to the method in form of Input and what actually exists in the DB. <li>If any Difference found while doing comparison, the method gets informed about data change and the Merge Process stops at this point. <li>If no difference found, PartyMergeAudit object is created which stores the currently existing data about the survivor Party.( before party merge takes place) <li>Now, Party Merge is performed. The data corresponding to both survivor and victim gets merged and set to survivor. <li>In case error occurs while performing the above merge, the errors gets

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>added to the response . &lt;li&gt;Then, Merge of party accounts and party relationships are done and errors, if any recorded, get added to response &lt;li&gt;Now, Merge takes place in other modules corresponding to the parties and errors, if any, get added to the Response. &lt;li&gt;After all the merging events have been performed, and no error in the response: &lt;ul&gt; &lt;li&gt;An Entity PartyMergeHistory gets persisted into the DB, indicating the parties have successfully been merged. &lt;li&gt;Party Merge status in PartyMergeState is updated to COMPLETED. &lt;li&gt;The Entity Status of Victim party is set to CLOSED., indicating victim party no longer exists. &lt;li&gt;The party merge status of Victim party is set to MERGED, indicating the party has been merged to some other party. &lt;li&gt;post merge PartyMergeAudit is formed, which contains survivor party related data after party merge has successfully taken place. &lt;li&gt;This PartyMergeAudit Entity gets persisted into the DB &lt;li&gt;The results of the PartyMerge event gets persisted into the DB. &lt;/ul&gt; &lt;li&gt;otherwise, the Errors in response get persisted in the Party Merge State Table. &lt;/ul&gt; &lt;li&gt;Validate Party Merge (Virtual simulation of Party Merge) &lt;ul&gt; &lt;li&gt;Merge takes place in other modules corresponding to the parties. If any error, gets set to the Response. &lt;/ul&gt; &lt;/ul&gt; &lt;/ul&gt; @param sessionContext @param partyMergeDTO @param isActualMerge @return @throws FCRException @throws ClassNotFoundException @throws IOException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method is used for canceling a merge request. when a partyMergeRequest is raised, and if due to some reason it couldn't succeed then the related parties should get rolled back to their original states: <ul style="list-style-type: none"> <li>Victim Party Status should get marked back to <b>ACTIVE</b> from previous status</li> <li><b>MERGE_INITIATED</b>. Party Merge Status for PartyMergeState should get marked to <b>CANCELLED</b> from previous status</li> <li><b>INITIATED</b></li> </ul>
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the status in the workflow tracker summarized entity. @param mergeId @param partyMergeStatus
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	onboardPartyThroughProcess	This method will call the create customer process to onboard a new party. This method will be used when a record for a party existing in other systems such as OCH needs to be created within OBP. @param sessionContext @param partyDataDTO @return @throws FatalException	This service will onboard a OCH only party and create a record within OBP system. @param sessionContext @param String, externalPartyId, Indicates the Id used for inquiring details of the party that is to be onboarded from OCH. @param PartyType, partyType, Indicates the party type for the party being onboarded. @return PartyOnBoardingResponse @throws FatalException
IPartyWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyWorkflowAdapter	onboardPartyThroughProcess	This method will call the create customer process to onboard a new party. This method will be used when a record for a party existing in other systems such as OCH needs to be created within OBP. @param sessionContext @param partyDataDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPCAdapter	inquireChequeStatus	This service fetches the cheque status details @param sessionContext @param chequeStatusInquiryDTO	
IDSLPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPCAdapter	issue	Issue () method of BankDraftApplicationService for issuing the bank draft .  It accepts 3 parameters-----   <ul> <li>BankDraftDTO - BankDraftDTO has all the information related to bankDraft (Settlement mode, Amount, serial-number, etc.).  <li>SessionContext - Standard header information for the application.   <li>MessageType </ul>This method will return the bank draft response for successful issuance of bank draft.   It will have the BankDraft serial no for further reference and tracking.   If an instrument is liquidated over the counter, then the serial number is used to track the instrument uniquely.   The serial number comprises of branch code followed by issue date followed by running serial number. @see com.ofss.fc.app.pc.bankdraft.service .IBankDraftApplicationService#issue(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO, java.lang.Object)	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPCAdapter	resolvePayableAtInstitution	This service represents the resolution of payable at institution @param sessionContext @return @throws FatalException	
IDSLPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPCAdapter	resolvePayableAtInstitution	This service represents the resolution of payable at institution @param sessionContext @return @throws FatalException	
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		The method is used to create CCI policy for each applicant inside product group. The CCI policy will be created   only if CCI is applicable at the product group level. This will call the Consumer Credit Insurance Interface   which will in-turn make a call to the corresponding CCI provider to actually create the CCI policy.  @fcb.param MI SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @fcb.param MI LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, ConsumerCreditInsuranceDetailDTO, etc 
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		The method is used to create CCI policy for each applicant inside product group. The CCI policy will be created   only if CCI is applicable at the product group level. This will call the Consumer Credit Insurance Interface   which will in-turn make a call to the corresponding CCI provider to actually create the CCI policy.  @fcb.param MI SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @fcb.param MI LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, ConsumerCreditInsuranceDetailDTO, etc 
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		This method is used to create CCI policy linkages for applicant applied for Consumer Credit Insurance in specific product group. @param sessionContext, It contains contain related to User, Bank Code , Branch etc. @param productGroupLinkageDTO, Represents details to specific product group. @return CCIDetailsResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicy		This method will initiate CCI Policy Creation at Application Level. This will be directly called from SOA Process.     It will internally call CCI Policy Creation at PG Level similar to one from Human task at PG level initiation. Based on final policy status by combining policy status of all PGs in Application further process takes place.     @param sessionContext @param applicationId
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicyWithQuote		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicyWithQuote		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createCCIPolicyWithQuote		This method is used to create CCI policy linkages for applicant applied for Consumer Credit Insurance in specific product group. @param sessionContext, It contains contain related to User, Bank Code , Branch etc. @param productGroupLinkageDTO, Represents details to specific product group. @return CCIDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This adapter call is used to fetch CCI premium amount quote.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This adapter call is used to fetch CCI premium amount quote.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		The method is called when the user wants to know if CCI is applicable for any product group in an application. This method will be called through BPEL process to check whether human task should be created for Consumer Credit Insurance or not. Inquiry of CCI policies for each applicant will be done and if Policy is rejected for all the applicants, then it will return NOT APPLICABLE. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, Indicates the application Id @return CCIApplicableResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		The method is called on submit of Consumer Credit Insurance Task. There are basically three scenarios that may happen on submission of CCI task. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, Indicates the application Id @return TransactionStatus @throws FatalException @exception OverridablePolicyViolationException FC_OR_CCI_0002:  This exception will be throws even if one product group inside an Application has CCI policy status as OR_CCI_POLICY_NOT_PROCESSED{@link OriginationErrorConstants.OR_CCI_POLICY_NOT_PROCESSED} .
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This method is used to fetch all policy details related to Consumer Credit Insurance for each applicant in given product group. This method will be called from Human Task on click of view details. @param sessionContext, It contains contain related to User, Bank Code , Branch etc. @param lendingProductGroupLinkageDTO, Represents details to specific product group. @return CCIPolicyDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This method will be called from UI to fetch Basic CCI Details for Applicant to be shown in CCI Human task as grid. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @throws FatalException @{@link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This will be called from back to Structure solution to cancel just the latest insurance policy request sent. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @param ignoreCCIRejection, this flag denotes whether cci policy rejected flag should be updated to true or not. this flag will come true for Stage "Back to structure solution" @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIPolicyDetails		This will fetch all CCI details available under given application. @param sessionContext, contains user information. @param applicationId, denotes the application ID @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateId		This adapter call is used to fetch CCI premium amount quote.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateId		This adapter call is used to fetch CCI premium amount quote.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateId		This method is used to create CCI policy linkages for applicant applied for Consumer Credit Insurance in specific product group. @param sessionContext, It contains contain related to User, Bank Code , Branch etc. @param productGroupLinkageDTO, Represents details to specific product group. @return CCIDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateId		This method is called from create lending application to persist the CCI quote details. At Funds table in application form, we generate an insurance id, fetch the CCI quote and persist the quote-facility linkage in insurance. The insurance Id is stored in CCIDetailDTO. On submit, if insurance id is present, we delete the existing quote-facility linkage and add quote-facility and quote-party linkages. If insurance id is not present, we generate the insurance id and create insurance-facility and insurance-party linkage. There would be no CCI quote data in this case. @param sessionContext @param productGroupLinkageDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This adapter call is used to fetch basic insurance details for given policy code.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This adapter call is used to fetch CCI premium amount quote.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This method will create InsuranceDTO from CCIDetailDTO @param cciDetail @param lendingPG @return
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This method would perform the following operations  <ul> <li>Derive the product group linkages via submission-helper</li> <li>Extract the lending product group linkages</li> <li>Update the CCI details by invoking the fetchInsurancePolicyBasicDetails() via Insurance adapter</li> <li>Update the CCI Details Dto in the applicant product group ref instance</li> </ul> @param sessionContext, Session context, needed to invoke the Insurance adapter @param submissionInputDTO, Submission Input Dto which is to be refreshed with the latest CCI details DTO @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This adapter call is used to fetch basic insurance details for given policy code.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This adapter call is used to fetch CCI premium amount quote.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This method will create InsuranceDTO from CCIDetailDTO @param cciDetail @param lendingPG @return
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This method will be called from UI to fetch Basic CCI Details for Applicant to be shown in CCI Human task as grid. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @throws FatalException {@link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsurancePolicyBasicDetails		This will fetch all CCI details available under given application. @param sessionContext, contains user information. @param applicationId, denotes the application ID @throws FatalException {@link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchCCIQuote		This method is called from funds table to fetch the CCI quote for all parties under a facility who have opted for CCI. @param sessionContext @param submissionId @param applicantDataDTOs @param productGroupLinkageDTO @param collateralDTOs @param creditFacilityDTOs @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelCCIPolicy		Adapter call to cancel insurance request or cci policy depending on where call is made from either from : <ol style="list-style-type: none"> <li>Back to structure solution.</li> <li>Edit/Withdraw Submission.</li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelCCIPolicy		Adapter call to cancel insurance request or cci policy depending on where call is made from either from : <ol> <li>Back to structure solution.</li> <li>Edit/Withdraw Submission.</li> </ol>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelCCIPolicy		This cancel will be called from edit/withdraw submission. We need to cancel all the proposed policies linked to the facility/party. @param facilityId @param partyId @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelCCIPolicy		This will be called from back to Structure solution to cancel just the latest insurance policy request sent. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @param ignoreCCIRejection, this flag denotes whether cci policy rejected flag should be updated to true or not. ,this flag will come true for Stage "Back to structure solution" @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	validateCCIPolicyForApplicants		Adapter call to insurance module to check where applicant is eligible for CCI.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	validateCCIPolicyForApplicants		Adapter call to insurance module to check where applicant is eligible for CCI.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	validateCCI PolicyForApplicants		This method will validate if applicant has applied for insurance. @param sessionContext, contains user information. @param applicantDataDTO, denotes the applicant for validation @param productGroupLinkageDTO, denotes productGroupLinkageDTO @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceLinkageByLinkedEntityIdAndLinkedState		Adapter call to fetch all the proposed policies linked to the facility/party before making a call to cancel policy during edit or withdraw submission.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceLinkageByLinkedEntityIdAndLinkedState		Adapter call to fetch all the proposed policies linked to the facility/party before making a call to cancel policy during edit or withdraw submission.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceLinkageByLinkedEntityIdAndLinkedState		This cancel will be called from edit/withdraw submission. We need to cancel all the proposed policies linked to the facility/party. @param facilityId @param partyId @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPaymentAccountForExistingCCI	This service call will stamp the input account id (The Account which is nominated for fees/premium (com.ofss.fc.app.origination.dto.core.pg.OfferReferenceDTO.isNominatedForFeesAndPremium) ) as nominated account for premium payment of the CCI policy with the given insurance id.	This service call will stamp the input account id (The Account which is nominated for fees/premium (com.ofss.fc.app.origination.dto.core.pg.OfferReferenceDTO.isNominatedForFeesAndPremium) ) as nominated account for premium payment of the CCI policy with the given insurance id.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPaymentAccountForExistingCCI	This service call will stamp the input account id (The Account which is nominated for fees/premium (com.ofss.fc.app.Origination.dto.core.pg.OfferReferenceDTO.isNominatedForFeesAndPremium) ) as nominated account for premium payment of the CCI policy with the given insurance id.	This service call will stamp the input account id (The Account which is nominated for fees/premium (com.ofss.fc.app.Origination.dto.core.pg.OfferReferenceDTO.isNominatedForFeesAndPremium) ) as nominated account for premium payment of the CCI policy with the given insurance id.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPaymentAccountForExistingCCI	This service call will stamp the input account id (The Account which is nominated for fees/premium (com.ofss.fc.app.Origination.dto.core.pg.OfferReferenceDTO.isNominatedForFeesAndPremium) ) as nominated account for premium payment of the CCI policy with the given insurance id.	@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllInsurancePoliciesByType		<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt;</li> <li>&lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @param sessionContext @param insuranceType @param productGroupCode</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllInsurancePoliciesByType		<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt;</li> <li>&lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt;</li> </ul> <p>@param sessionContext @param insuranceType @param productGroupCode</p>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllInsurancePoliciesByType		<p>This method call will be used on Applicant Details Grid if applicant opts for CCI. This method will return all Policy Details maintained in Credit Policy for product Group Code. If there are no policies maintained in CreditPolicy , then all policies returned by Insurance would be set in response. @param sessionContext, It contains contain related to User, Bank Code , Branch etc. @param insuranceType, represent an attribute of Type DebtInsuranceType productGroupCode, identifier for productGroupCode @throws FatalException</p>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	updateInsurance		<p>This method is called from create lending application to persist the CCI quote details. At Funds table in application form, we generate an insurance id, fetch the CCI quote and persist the quote-facility linkage in insurance. The insurance Id is stored in CCIDetailDTO. On submit, if insurance id is present, we delete the existing quote-facility linkage and add quote-facility and quote-party linkages. If insurance id is not present, we generate the insurance id and create insurance-facility and insurance-party linkage. There would be no CCI quote data in this case. @param sessionContext @param productGroupLinkageDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	activateProposedFacilityCCILinkages		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	activateProposedFacilityCCILinkages		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	activateProposedFacilityCCILinkages		@param facilityId @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveCCIQuoteAndLinkages		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveCCIQuoteAndLinkages		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveCCIQuoteAndLinkages		This method is used to save CCI Quote Linkages to insurance module and save Insurance id generated for each applicant applying for CCI policy on origination side. @param sessionContext @param lendingProductGroupLinkageDTO
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveCCIQuoteAndLinkages		This method is called from create lending application to persist the CCI quote details. At Funds table in application form, we generate an insurance id, fetch the CCI quote and persist the quote-facility linkage in insurance. The insurance Id is stored in CCIDetailDTO. On submit, if insurance id is present, we delete the existing quote-facility linkage and add quote-facility and quote-party linkages. If insurance id is not present, we generate the insurance id and create insurance-facility and insurance-party linkage. There would be no CCI quote data in this case. @param sessionContext @param productGroupLinkageDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	deleteInsuranceLinkagesForInsurance		This will be called from back to Structure solution to cancel just the latest insurance policy request sent. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @param ignoreCCIRejection, this flag denotes whether cci policy rejected flag should be updated to true or not. ,this flag will come true for Stage "Back to structure solution" @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		@param sessionContext @param dto @param lmiAmount @param accountsOpened @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchAllPremiumDetailsByInsuranceId		This method is used to build FacilityApprovalDataDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchLMIQuote		<p>The method is called from the Funding Table in UI to inquire the LMI quote from the LMI provider.&lt;br&gt;The applicant details and collateral details are required in addition to LendingProductGroupLinkageDTO&lt;br&gt;To get the Quote. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO , This stores information&lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @param ApplicantDataDTO [] applicantDataDTOs , This contains the details of the applicants who &lt;br&gt; have applied for the product group. @param CollateralDTO [] collateralDTOs , This contains the collateral details linked to the product group. @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001 &lt;br&gt; This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchLMIQuote		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMIPolicy		<p>The method is used to create LMI policy for a product group. The LMI policy will be created only if LMI is applicable at the product group level. This will call the Mortgage Insurance adapter which will in-turn make a call to the corresponding LMI provider to actually create the LMI policy. Since the Policy Creation Response is an asynchronous callback, an update method has been provided in the application service to update the policy details. Until then the LMI status is set to 'Pending' in origination. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return LRMPolicyDetailsResponse response @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001 This exception is thrown when we try to create a LMI policy for a Product Group for which LMI is not applicable.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMIPolicy		<p>This method will initiate LMI Policy Creation at Application Level. This will be directly called from SOA Process. &lt;br&gt; &lt;br&gt; It will internally call LMI Policy Creation at PG Level similar to one from Human task at PG level initiation. Based on final policy status by combining policy status of all PGs in Application further process takes place. &lt;br&gt; &lt;br&gt; Following Policy status results in further process:</p> <ul style="list-style-type: none"> <li>&lt;b&gt;&lt;u&gt;PolicyStatusType.APPROVED&lt;/u&gt;&lt;/b&gt;: If policy status is APPROVED and none of business policy validations failed or overridable exceptions occur LMI task won't be created and further task in Origination workflow will be created.&lt;/li&gt; &lt;li&gt;&lt;b&gt;&lt;u&gt;PolicyStatusType.PENDING&lt;/u&gt;&lt;/b&gt;: If policy status is PENDING then as per mock strategy you need to manually provide status and final premium from mock screen and decide further scenario.&lt;/li&gt; &lt;li&gt;&lt;b&gt;&lt;u&gt;PolicyStatusType.REJECTED&lt;/u&gt;&lt;/b&gt;: If policy status is REJECTED or other than above two then LMI task will be created. In this case either you can withdraw or edit submission or go back to Structure Solution.&lt;/li&gt; &lt;br&gt; &lt;br&gt; @param sessionContext @param applicationId</li> </ul>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMIPolicy		<p>This method will initiate CCI Policy Creation at Application Level. This will be directly called from SOA Process. &lt;br&gt; &lt;br&gt; It will internally call CCI Policy Creation at PG Level similar to one from Human task at PG level initiation. Based on final policy status by combining policy status of all PGs in Application further process takes place. &lt;br&gt; &lt;br&gt; @param sessionContext @param applicationId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		This adapter call will fetch all insurance details for given insurance ID.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		The method is used to create LMI policy for a product group. The LMI policy will be created   only if LMI is applicable at the product group level. This will call the Mortgage Insurance Interface   which will in-turn make a call to the corresponding LMI provider to actually create the LMI policy.  @fcb.param MI SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @fcb.param MI LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc 
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		This adapter call will fetch all insurance details for given insurance ID.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		The method is used to create LMI policy for a product group. The LMI policy will be created   only if LMI is applicable at the product group level. This will call the Mortgage Insurance Interface   which will in-turn make a call to the corresponding LMI provider to actually create the LMI policy.  @fcb.param MI SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @fcb.param MI LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, LendingProductGroupLinkageDTO, etc 
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		The method is called when the user wants to view the basic details of the LMI policy given the ApplicationId.  @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, Indicates the application Id @return LRMPolicyDetailsResponse @exception OriginationErrorConstants.OR_INSURANCE_NOT_FOUND : FC_OR_MI_0002   This exception is thrown when we try to fetch the policy details   for an application for which policy has not been initiated.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		This method is used to fetch the LRM Policy Details using application id and product group serial no. This is used to populate the Mortgage Insurance Grid on the screen. If policy status is APPROVED, then Insurance Premium is fetched from Insurance. @param sessionContext @param lrmDetailsDTO @return LRMPolicyDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		This method is used to fetch LMI Policy Details for Given Product Group Details. @param sessionContext @param applicationId @param productGroupSerialNumber @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	fetchInsuranceDetails		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	addOrUpdatePremiumDetails		This method updates the premium and quote details. This method is to be used for any updates to be done to insurance record after policy is created @param sessionContext @param premiumQuoteDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	addOrUpdatePremiumDetails		The method is used to create LMI policy for a product group. The LMI policy will be created   only if LMI is applicable at the product group level. This will call the Mortgage Insurance adapter   which will in-turn make a call to the corresponding LMI provider to actually create the LMI policy.  Since the Policy Creation Response is an asynchronous callback, an update method has been provided   in the application service to update the policy details.  Until then the LMI status is set to 'Pending' in origination. @param SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc  @return LRMPolicyDetailsResponse response @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001   This exception is thrown when we try to create a LMI policy   for a Product Group for which LMI is not applicable.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	addOrUpdatePremiumDetails		LMI is not capitalized but with Negotiation. This method will compare LMI Premium and Negotiated LMI and nullify if latter is less. Also update the waiver amount if new premium is more than net borrower premium @param response @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPayAccountForExistingLMI		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPayAccountForExistingLMI		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	nominatePremiumPayAccountForExistingLMI		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveOrUpdateLMIWaiverDetails		LMI is not capitalized but with Negotiation. This method will compare LMI Premium and Negotiated LMI and nullify if latter is less. Also update the waiver amount if new premium is more than net borrower premium @param response @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveOrUpdateLMIWaiverDetails		This method will be called during submission creation through origination process. It will create Insurance, Premium Quote and LMI Waiver Details if captured in Insurance module. It will store LMI Details in Origination as well. @param context @param productGroupLinkageDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveOrUpdateLMIWaiverDetails		This method will be used to save or update the LMI Waiver Details captured as a part of Structure Solution in Post Assessment. @param sessionContext @param waiverDTO @return {@link TransactionStatus} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveOrUpdateLMIWaiverDetails		This method will mark LMI Negotiated or waived as a part of Structure Solution as LMIWaiverStatusType.WAIVED. And it will additionally raise an overridable exception for approval process. @param sessionContext @param applicationId @param productGroupSerialNumber @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	saveOrUpdateLMIWaiverDetails		This API is used to create or update LMI Policy Linked to the submission and the productGroup. It performs the following operations <ol> <li>fetches LMI Details For PG</li> <li>IF LMI Details not present then <ol> <li>saves LMI Quote and Linkages</li></ol> </li> <li>else <ol> <li>update LRM details</li> <li>create LMI</li> <li>saves or updates LMI Waiver Details</li> </ol> </li> </ol> @param sessionContext @param productGroupLinkageDTO @param submissionId @return LRMPolicyDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		This method will be unique identifier stored in origination as well as insurance side.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		This method will be unique identifier stored in origination as well as insurance side.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		This method will be called during submission creation through origination process. It will create Insurance, Premium Quote and LMI Waiver Details if captured in Insurance module. It will store LMI Details in Origination as well. @param context @param productGroupLinkageDTO @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		This method will be used to save or update the LMI Waiver Details captured as a part of Structure Solution in Post Assessment. @param sessionContext @param waiverDTO @return {@link TransactionStatus} @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	generateInsuranceId		This API is used to create or update LMI Policy Linked to the submission and the productGroup. It performs the following operations <ol> <li>fetches LMI Details For PG</li> <li>IF LMI Details not present then <ol> <li>saves LMI Quote and Linkages</li></ol> </li> <li>else <ol> <li>update LRM details</li> <li>create LMI</li> <li>saves or updates LMI Waiver Details</li> </ol> </li> </ol> @param sessionContext @param productGroupLinkageDTO @param submissionId @return LRMPolicyDetailsResponse @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMI		The method is used to create LMI policy for a product group. The LMI policy will be created   only if LMI is applicable at the product group level. This will call the Mortgage Insurance adapter   which will in-turn make a call to the corresponding LMI provider to actually create the LMI policy.  Since the Policy Creation Response is an asynchronous callback, an update method has been provided   in the application service to update the policy details.  Until then the LMI status is set to 'Pending' in origination. @param SessionContext, sessionContext Session Context which contains information   such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO, This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc  @return LRMPolicyDetailsResponse response @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001   This exception is thrown when we try to create a LMI policy   for a Product Group for which LMI is not applicable.
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMI		This method will be called during submission creation through origination process. It will create Insurance, Premium Quote and LMI Waiver Details if captured in Insurance module. It will store LMI Details in Origination as well. @param context @param productGroupLinkageDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMI		<p>This method will initiate LMI Policy Creation at Application Level. This will be directly called from SOA Process. &lt;br&gt; &lt;br&gt; It will internally call LMI Policy Creation at PG Level similar to one from Human task at PG level initiation. Based on final policy status by combining policy status of all PGs in Application further process takes place. &lt;br&gt; &lt;br&gt; Following Policy status results in further process:</p> <ul style="list-style-type: none"> <li>&lt;b&gt;&lt;u&gt;PolicyStatusType.APPROVED&lt;/u&gt;&lt;/b&gt;: If policy status is APPROVED and none of business policy validations failed or overridable exceptions occur LMI task won't be created and further task in Origination workflow will be created.&lt;/li&gt; &lt;li&gt;&lt;b&gt;&lt;u&gt;PolicyStatusType.PENDING&lt;/u&gt;&lt;/b&gt;: If policy status is PENDING then as per mock strategy you need to manually provide status and final premium from mock screen and decide further scenario.&lt;/li&gt; &lt;li&gt;&lt;b&gt;&lt;u&gt;PolicyStatusType.REJECTED&lt;/u&gt;&lt;/b&gt;: If policy status is REJECTED or other than above two then LMI task will be created. In this case either you can withdraw or edit submission or go back to Structure Solution.&lt;/li&gt; &lt;br&gt; &lt;br&gt; @param sessionContext @param applicationId</li> </ul>
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMI		<p>This API is used to create or update LMI Policy Linked to the submission and the productGroup. It performs the following operations &lt;ol&gt; &lt;li&gt;fetches LMI Details For PG&lt;/li&gt; &lt;li&gt;IF LMI Details not present then &lt;ol&gt; &lt;li&gt;saves LMI Quote and Linkages&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;else &lt;ol&gt; &lt;li&gt;update LRM details&lt;/li&gt; &lt;li&gt;create LMI&lt;/li&gt; &lt;li&gt;saves or updates LMI Waiver Details&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;/ol&gt;</p> <p>@param sessionContext @param productGroupLinkageDTO @param submissionId @return LRMPolicyDetailsResponse @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	createLMI		This method will initiate CCI Policy Creation at Application Level. This will be directly called from SOA Process.     It will internally call CCI Policy Creation at PG Level similar to one from Human task at PG level initiation. Based on final policy status by combining policy status of all PGs in Application further process takes place.     @param sessionContext @param applicationId
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		This method will fetch waiver Details for given insurance identifier. If there are no waiver details found, it will indicate that no negotiation was done for Customer LMI Premium. @param sessionContext @param insuranceId @return
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		This method will mark LMI Negotiated or waived as a part of Structure Solution as LMIWaiverStatusType.WAIVED. And it will additionally raise an overridable exception for approval process. @param sessionContext @param applicationId @param productGroupSerialNumber @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		This method will check if there was any negotiation or waiver done for LMI Premium. @param sessionContext @param applicationId @param productGroupSerialNumber @return @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		@param sessionContext @param dto @param lmiAmount @param accountsOpened @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	inquireLMIWaiverDetailsDTO		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	deleteLMIWaiverDetails		This method is used to delete the LRM waiver details. This method will be called only in Post Assessment stage during Structure Solution. @param SessionContext, sessionContext Session Context, which contains information such as user id, branch, branch code, channel etc. @param LenderRiskMgmtWaiverDetailsDTO IrmWaiverDetailsDTO, Stores the LRM Policy waiver details @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		This method cancels all LMI policies for the given Facility identified by facilityId. @param sessionContext @param facilityId @return TransactionStatus @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		This method will called when from any stage above MortgageInsurance in application is reverted back to structure solution or edit submission. In this stage policy will be cancelled corresponding status for policy will be updated in Insurance Module. @param sessionContext @param applicationId @param productGroupSerialNumber @param isEditSubmission
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		This method LMI for given application ID and pg serial number which corresponds to Product Group Level removal of LMI Details. @param sessionContext @param applicationId @param productGroupSerialNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		This API is used to create or update LMI Policy Linked to the submission and the productGroup. It performs the following operations <ol style="list-style-type: none"> <li>fetches LMI Details For PG</li> <li>IF LMI Details not present then <ol style="list-style-type: none"> <li>saves LMI Quote and Linkages</li> <li>update LRM details</li> <li>create LMI</li> <li>saves or updates LMI Waiver Details</li> </ol> </li> </ol> @param sessionContext @param productGroupLinkageDTO @param submissionId @return LRMPolicyDetailsResponse @throws FatalException
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	cancelLMIPolicy		This method will first cancel the Mortgage Insurance and delete the record. @param sessionContext @param applicationId
IOriginationInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationInsuranceAdapter	activateProposedInsuranceLinkages		@param sessionContext @param applicationId @param prodGrpLinkage @param lcmAdapter @return @throws FatalException
IAgentLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This service will return AccountSummaryDTO for all the account under facilityId provided in input. @param facilityId :facility id of which all child Account will be returned back @return AccountSummaryResponse
IAgentLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method is responsible for fetching the account. @param accountId @return AccountDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerExternalAccountsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerExternalAccountsAdapter	fetchExternalAccounts		@return Object
IHardshipReliefAssistanceApplyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IHardshipReliefAssistanceApplyAdapter	save	@param hardshipReliefRequestDTO @return TransactionStatus @throws FatalException	@param hardshipReliefRequestDTO @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	@param sweepInInstructionDTO @return
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	@param sweepInInstructionDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	fetchBillDetails	This method fetches the basic biller details. @param instructionKeyDTO @return	
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	fetchBillDetails	This method fetches the basic biller details. @param instructionKeyDTO @return	@param fundsManagementDTO @return
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	checkPendingTransactions	This method is used to check whether the Account has pending Transactions or not. It is being accessed during Account Closure. @param accountId @param moduleType @throws FatalException	<p>TODO:ATM card needs to be marked as destroyed. Amit to get back where this status is maintained. This method will be used for the final closure of the account. It will settle the interest components, fees, tax, process the closure withdrawal and update the account status to Closed. The method works as follows :</p> <ul style="list-style-type: none"> <li>Validate whether the account details are permissible for account closure or not by calling the domain service {@link DDAClosureService#validateAccountClosure}. The value of the parameter isAccountBeingClosed will be passed as True.</li> <li>Call the method {@link DDAClosureService#processFeePayment} to post the fee entries due for the account and update the account balance</li> <li>Call the method {@link DDAClosureService#processInterestAndTaxSettlement} to compute the interest and pass accrual and capitalization entries.</li> <li>Call the method {@link DDAClosureService#markUnpaidChequesAsDestroyed} to update the status of all the unused cheques, so that they cannot be issued by the account holder once the account is closed.</li> <li>If the input.closureRequestId is null, it implies that the screen is being used standalone Call the method {@link DemandDepositAccountService#modifyClosureReason}</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ns) to update the closure reasons. Call the method <a href="#">{@link IDDASettlementAdapter#executePreferenceOnline}</a> with the settlementAmount returned by the service <a href="#">{@link DDAClosureService#validateAccountClosure}</a> to pay the credit amount to the customer in the mode chosen by him. Else Call the service <a href="#">{@link DDAClosureService#fetchClosureRequest}</a> to inquire the request details and retrieve the settlementReferenceNumber.</p> <ul style="list-style-type: none"> <li>Call the method <a href="#">{@link IDDASettlementAdapter#processSettlement}</a> with the settlementAmount returned by the service <a href="#">{@link DDAClosureService#validateAccountClosure}</a> to pay the credit amount to the customer in the mode chosen by him.</li> <li>Delete the future dated events for the account by calling the service <a href="#">{@link ActionLogger#removeAllIDDAActions}</a></li> <li>Unregister the pricing plan linked to the account by calling the service <a href="#">{@link IDDA PricingAdapter#closePricingPlan }</a></li> <li>Unregister the interest plan linked to the account by calling the service <a href="#">{@link IDDAInterestAdapter#unregisterPlan }</a></li> <li>Finally call the method <a href="#">{@link AccountStatusManagementService#modifyAccountStatus}</a> to mark the account as closed.</li> <li>Check the overridable conditions by calling the method <a href="#">{@link OverridableFinancialBusinessPolicy#validate}</a></li> </ul> <p>@param accountId @param closureDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	validateChequeStalePeriod	This method would validate whether the cheque is in stale period by taking cheque number,date,acct no as input	This method would be used as wrapper service for DashBoard services, this method would first check the cheque number, cheque stale period by calling payments services and then would call transfer funds method @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI, String, initiatorAccountId, 1..40, accountId of the account which is to be debited. @fcb.param MI, String[], beneficiaryAccountId, 1..40, accountId's which is to be credited. @fcb.param I, String, description, , description/comments for the fund transfer transaction. @fcb.param MI, MoneyDTO, transactionAmount, , the amount to be debited. @fcb.param MI, String, chequeNumber, , cheque number to be used in funds transfer. @fcb.param MI, Date, chequeDate, , date on which cheque was drawn to be used in funds transfer. @fcb.param MI, String, payeeName, , name of the payee. @return @throws FatalException
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	payUtilityBill	This method is used to do the bill payment for one time only. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	This method is used to make a one time bill payment when request made from the dashboard. this method takes minimal input in billPaymentRequestDTO to perform one time bill payment @param billPaymentRequestDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAadapter	payUtilityBill	This method is used to do the bill payment for one time only. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	This method initializes the Bill Payment. It takes Bill Transaction DTO as an input parameter. The bills payments module will have data per account and biller combination for all the transactions hitting the account. Irrespective of the status of the transaction, meaning whether it was successful or failure, the data will be stored and displayed. Based on the filter options, the data will be available. Utility payments can be initiated either through Internet banking channel by the customer or can be done at the branch. This document covers the scope of the branch banking only. Customer can initiate bill payments to a particular biller only after adding the biller as registered. The user can opt to make payments either through a) Own Account Debit b) Cheque c) Cash d) Credit Cards In each case user captures specific payment details based on the payment mode. For payments through Cheque the processing will be possible only after cheque realization and till that time the transaction will not be executed. @param sessionContext @param billPayDTO @param mode @throws FatalException
IDDAPCAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAadapter	payUtilityBill	This method is used to do the bill payment for one time only. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for one time bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException
IDDAPCAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAadapter	payUtilityBill	This method is used to do the bill payment for one time only. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for one time bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	setUpPeriodicBillpayment	This method is used to do the periodic bill payment setup. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	This method is used to make a set up for periodic bill payment when request made from the dashboard. this method takes minimal input in billPaymentRequestDTO to perform periodic bill payment setup @param billPaymentRequestDTO @throws FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	setUpPeriodicBillpayment	This method is used to do the periodic bill payment setup. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for set up pf periodic bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	setUpPeriodicBillpayment	This method is used to do the periodic bill payment setup. it will take fundTransferRequestDTO as input @param fundTransferRequestDTO @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for set up pf periodic bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeDetails		This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	inquirePayeeDetails		
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchByCounterpartyAccNo	This method is used for fetching counter party details. It will take input as partyId of initiator account and counter party account number and return counterpartyDTO @param partyId @param counterpartyAccNo @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchByCounterpartyAccNo	This method is used for fetching counter party details. It will take input as partyId of initiator account and counter party account number and return counterpartyDTO @param partyId @param counterpartyAccNo @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchByCounterpartyAccNo	This method is used for fetching counter party details. It will take input as partyId of initiator account and counter party account number and return counterpartyDTO @param partyId @param counterpartyAccNo @return	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchByCounterpartyAccNo	This method is used for fetching counter party details. It will take input as partyId of initiator account and counter party account number and return counterpartyDTO @param partyId @param counterpartyAccNo @return	{@link IDDAPCAdapter#fetchByCounterpartyAccNo(String, String)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	transferFundstoExternalAccount	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as FundExchangeRequestDTO and return CommonFundExchangeResponse. @param fundExchangeRequestDTO @return	This method will mark a cheque paid on initiator CASA account. Initiate an outgoing payment request of the same amount to an external account. This method would be used as wrapper service for com.ofss.fc.app.pc.core.spi.FundExchangeService.execute(SessionContext, FundExchangeRequestDTO). @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI, String, initiatorAccountId, 1..40, accountId of the account which is to be debited. @fcb.param MI, String, initiatorPartyId, 1..40, partyId of the account which has initiated the transaction. @fcb.param MI, String, counterpartyId, 1..40, counterparty Id to which the funds will be transferred. @fcb.param I, String, description, , description/comments for the fund transfer transaction. @fcb.param MI, MoneyDTO, transactionAmount, , the amount to be debited. @fcb.param MI, String, chequeNumber, , cheque number to be used in funds transfer. @fcb.param MI, Date, chequeDate, , date on which cheque was drawn to be used in funds transfer. @return @throws FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	transferFundstoExternalAccount	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as FundExchangeRequestDTO and return CommonFundExchangeResponse. @param fundExchangeRequestDTO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchCounterparty	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as CounterpartyDTO and return CounterpartyDTO. @param counterpartyDTO @return	{@link IDDAPCAdapter#fetchAllCounterParty(CounterpartyDTO)}
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchCounterparty	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as CounterpartyDTO and return CounterpartyDTO. @param counterpartyDTO @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchCounterparty	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as CounterpartyDTO and return CounterpartyDTO. @param counterpartyDTO @return	This method will mark a cheque paid on initiator CASA account. Initiate an outgoing payment request of the same amount to an external account. This method would be used as wrapper service for com.ofss.fc.app.pc.core.spi.FundExchangeService.execute(SessionContext, FundExchangeRequestDTO). @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI, String, initiatorAccountId, 1..40, accountId of the account which is to be debited. @fcb.param MI, String, initiatorPartyId, 1..40, partyId of the account which has initiated the transaction. @fcb.param MI, String, counterpartyId, 1..40, counterparty Id to which the funds will be transferred. @fcb.param I, String, description, , description/comments for the fund transfer transaction. @fcb.param MI, MoneyDTO, transactionAmount, , the amount to be debited. @fcb.param MI, String, chequeNumber, , cheque number to be used in funds transfer. @fcb.param MI,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Date, chequeDate, , date on which cheque was drawn to be used in funds transfer. @return @throws FatalException
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	fetchCounterparty	This method will execute an outgoing payment request for initiatorAccount to an external account. it use input as CounterpartyDTO and return CounterpartyDTO. @param counterpartyDTO @return	
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	simulateTransactionDetailsForAccountNoAfterTheGivenDate	This method is used to fetch the pc values dates for the given account id between the current date and the end date @param accountId @param startDate @return	This method is used to check whether there is pc clearance date after the simulation date and if there is AUF facility linked to the account
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	simulateTransactionDetailsForAccountNoAfterTheGivenDate	This method is used to fetch the pc values dates for the given account id between the current date and the end date @param accountId @param startDate @return	This method is used to fetch the pc values dates for the given account id between the current date and the end date
IDDAPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPCAdapter	simulateTransactionDetailsForAccountNoAfterTheGivenDate	This method is used to fetch the pc values dates for the given account id between the current date and the end date @param accountId @param startDate @return	This method is used to check whether if the cheque value dates is after the simulation date entered.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	checkIfAccountIsMandatoryInBundle	This method is used to check whether the account is nominated as mandatory account in the offer bundle setup. @param accountId @param offerId TODO @param bundleReferenceId TODO @return	This is the public method exposed in the class. It internally calls the private methods to perform theValidations. The private methods called are : <ol><li>validateAccountLinkages</li><li>validateAccountFacilities</li><li>validateAccountBalances</li><li>validateStatus</li></ol>
IDDAPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPCAAdapter	checkIfAccountIsMandatoryInBundle	This method is used to check whether the account is nominated as mandatory account in the offer bundle setup. @param accountId @param offerId TODO @param bundleReferenceId TODO @return	To check the account is mandatory in bundle set up. Because if this is the mandatory account and if this account is closed, the other accounts under that bundle will stop receiving benefits from the bundle.
IInsurancePartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsurancePartyAdapter	fetchPartyBasicInformation		
IInsurancePartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsurancePartyAdapter	fetchPartyDemographicDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementReversalAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementReversalAdapter	reverseSettlementInstruction	This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType TODO @param narration TODO @param context Indicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException	
ILoanExternalOffsetAccountApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanExternalOffsetAccountApplicationAdapter	insertOffsetAccountBalance	This method is used to store the account balances for the external account linked to loan account for principal offset. This will be called by LoanExternalOffsetAccountApplication which is called by GEFU processor. @param applicationContext @param accountId @param currencyCode @param offsetAccountBalance @param effectiveDate @throws FatalException	This method is used to store the account balances for the external account linked to loan account for principal offset. @param applicationContext @param externalOffsetAccountDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanExternalOffsetAccountApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanExternalOffsetAccountApplicationAdapter	linkExternalAccountForOffset	This method is used to link the external account to loan account for principal offset. It stores all the necessary information of the external account linked to the loan account. @param applicationContext @param loanAccountId @param offsetAccountId @param offsetAccountProductCode @param productCode @param processDate @param linkageStartDate @param linkageExpiryDate @param offsetAccountCurrencyCode @param offsetAccountCustomerName @param offsetAccountHolders @throws FatalException	This method is used to link the external account to loan account for principal offset. It stores all the necessary information of the external account linked to the loan account. @param applicationContext @param loanAccountId @param offsetAccountId @param offsetAccountProductCode @param productCode @param processDate @param linkageStartDate @param linkageExpiryDate @param offsetAccountCurrencyCode @param offsetAccountCustomerName @param offsetAccountHolders @throws FatalException
ILoanExternalOffsetAccountApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanExternalOffsetAccountApplicationAdapter	storeExternalOffsetAccountBalance	This method is used to store the account balances for the external account linked to loan account for principal offset. @param applicationContext @param offsetAccountId @param offsetAccountBalance @param processDate @param offsetAccountCurrencyCode @throws FatalException	This method is used to store the account balances for the external account linked to loan account for principal offset. @param applicationContext @param offsetAccountId @param offsetAccountBalance @param processDate @param offsetAccountCurrencyCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchProcessAdapter	initiateCurrencyIndentProcess		<p>This method is used to save an Indent that is been raised by any branch. This method creates an entry of indent that is been raised in the currency indent table and also a record in currency indent history table for audit. This method also initiates currency indent BPEL process.</p> <p>@fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param  MI,CurrencyIndentDTO,currencyIndentDTO, ,currencyIndentDTO @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IBranchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchProcessAdapter	initiateCurrencyIndentProcess		<p>Once an indent is been raised all the updation and changes or maintaining approval or rejection of currency indent is been done by this method. This method will update any changes made on a raised indent in currency indent table and changed record is been added to currency indent history table for auditing. If the indent is been updated from NextGen screen before any action on it is been taken by the approver in workflow, the old process is been terminated and a new process is been initiated.</p> <p>@fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param  MI,CurrencyIndentDTO,currencyIndentDTO, ,currencyIndentDTO @return {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchProcessAdapter	initiateCurrencyIndentProcess		
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardDMSAdapter	getInboundDocumentChecklist	This service will creates and fetches the list of inbound documents grouped under a checklist. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service creates the inbound document checklist for the add-on card holder  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,AddOnCardHolderDTO, addOnCardHolderDTO, ,This class holds the information of add on card  holder for which inbound document checklist is to be created  @fcb.param CreditCardDetailsDTO, creditCardDetailsDTO, ,This class holds the information of credit card  to which add-on card holder is being linked. This is optional argument when add-on card holder is linked  during origination of credit card. In such case creditCardAccountConfigurationDTO parameter is required   @fcb.param CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, ,This class holds the information of credit card  account configuration. This is optional argument when add-on card holder is added from credit card servicing screen  However in such case, creditCardDetailsDTO parameter is required  @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException @exception   {@link CardErrorConstants#INBOUND_DOC_CHKLIST_FAILED} : OBP_CC_0007   This exception is thrown if there is a problem in inbound document checklist generation.    
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardDMSAdapter	getInboundDocumentChecklist	This service will creates and fetches the list of inbound documents grouped under a checklist. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This method is used to update document list for add on new parties @param sessionContext @param addOnCardHolderDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardDMSAdapter	getInboundDocumentChecklist	This service will creates and fetches the list of inbound documents grouped under a checklist. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO,, Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service will fetch the list of inbound documents grouped under a checklist.  @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO,, Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardDMSAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardDMSAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardDMSAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This method is used to update document list for add on new parties @param sessionContext @param addOnCardHolderDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardDMSAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This service will update the checklist information. For applicants that are new, applicant serial number is initially used to create inbound document checklist. Once the applicants are on boarded, the party identifiers generated are updated back to the checklist. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistDTO, documentChecklistDTO, , Indicates the document checklist information for update. It contains the party identifiers updated in the checklist documents. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICardDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardDMSAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsForAccountUpdate	This method intimates Account related information to Collection @param accountFeederWrapperDTO	
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsForAccountUpdate	This method intimates Account related information to Collection @param accountFeederWrapperDTO	Method to be called for intimation of DPD to Collections after the Asset Classification

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsForAccountUpdate	This method intimates Account related information to Collection @param accountFeederWrapperDTO	Method to be called for intimation updated Provision amounts(currently SP) to Collections
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsForAccountUpdate	This method intimates Account related information to Collection @param accountFeederWrapperDTO	
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsforPayments	This method intimates account payment related information to Collection @param feederPaymentsWrapperDTO	
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsforPayments	This method intimates account payment related information to Collection @param feederPaymentsWrapperDTO	Method to be called for intimation on recovery of a Written-off account.
IAssetClassificationCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationCollectionAdapter	updateCollectionsforPayments	This method intimates account payment related information to Collection @param feederPaymentsWrapperDTO	
IAbstractDomainRepriceDerivationStrategyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IAbstractDomainRepriceDerivationStrategyAdapter	fetchPricesForDomainChange		Returns the List the arrangements effected as a result of domain Fact Change. @param sessionContext @param changeEvent @return array of {@link AccountRepriceDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will trigger reverse accounting for the CASA account for which business payment or collection was initiated, in case an error is encountered during procurement or disbursement of funds.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventAmountTagDetailsDTOArray <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fileType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will raise suspense accounting when reversal transaction to the beneficiary fails. @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fileType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, String, ledgerCode, Indicates the intermediary hold ledger for business transfer transaction. @fcb.param MI, boolean, isFirstCall <li>Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.</li> @throws BusinessException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. @fcb.param transactionLog @fcb.param accountingEventAmountTagDetailsDTOArray @throws FatalException @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will raise VDC specific accounting. @fcb.param context @fcb.param transactionLog @fcb.param messageLogItem @fcb.param notificationDTO @fcb.param eventId TODO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	@param transactionContext @param transactionLog @param amountTagDetailsDTO @param messageLogItem @param finalTagDTO @param accountingEventAmountTagDetailsDTO @param eventId @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	@param transactionContext @param transactionLog @param amountTagDetailsDTO @param messageLogItem @param sequenceNumber @param finalTagDTO @param accountingEventAmountTagDetailsDTO @param eventId @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	{@link OutgoingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense @see com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.entity .log.TransactionLog)
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method builds the accounting leg and triggers accounting for Business Collections.   @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param messageLogItem @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will be used to approve the transaction. When the transaction will get committed via 2FA delay then by using approve method from the work-list one can approve the transaction. Approve method will delegate call to this method. This method will fetch the transaction log based on the work-item id and mark the message state as READY_FOR_DISPATCH. @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will be used to reject the transaction. When the transaction will get committed by marking 2FA delay and when one called reject method from the work-list call will delegate to this method. This method will call the reverse method of the respective service. That means it is going to reverse the transaction which is committed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Debit&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_DEBIT&lt;/td&gt; &lt;td&gt;C44&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E20&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT&lt;/td&gt; &lt;td&gt;E21&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E22&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Pass to Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_PASS_TO_REJECT&lt;/td&gt; &lt;td&gt;E23&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C45&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject to Pass&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;C46&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E24&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;O/w Clearing for I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_OUTWARD_INWARD_REJECTS&lt;/td&gt; &lt;td&gt;E25&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E26&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Float Extension&lt;/td&gt; &lt;td&gt;CL_CHQ_FLOAT_EXTENSION&lt;/td&gt; &lt;td&gt;E27&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E28&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do the accounting for CASA credit process and GL debit.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO[], amountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method raises an accounting event (specially created for verification) for the payment made to the counter party bank. @param accountingEventAmountTagDetailsDTOArray @param transactionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<p>This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response.</p> <pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Deposit&lt;/td&gt; &lt;td&gt;CL_CHQ_DEPOSIT_TO_CASA&lt;/td&gt; &lt;td&gt;E01&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E02&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Immediate Credit (only to CASA accounts)&lt;/td&gt; &lt;td&gt;CL_CHQ_IMM_CREDIT_TO_CASA&lt;/td&gt; &lt;td&gt;E03&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;C63&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation with partial immediate credit&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_PARTIAL_IMM_CR&lt;/td&gt; &lt;td&gt;E04&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;BALANCE_CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C40&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;BALANCE_CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation without immediate credit&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_NO_IMM_CREDIT&lt;/td&gt; &lt;td&gt;E05&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C41&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to TD account&lt;/td&gt; &lt;td&gt;CL_CHQ_REALiZE_TO_TD&lt;/td&gt; &lt;td&gt;E06&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;T64&lt;/td&gt; &lt;td&gt;CLEARING_TDACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to Loans account&lt;/td&gt; &lt;td&gt;CL_CHQ_REALiZE_TO_LOAN&lt;/td&gt; &lt;td&gt;E07&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;L01&lt;/td&gt; &lt;td&gt;CLEARING_LOANACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to Biller&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_TO_BILLER&lt;/td&gt; &lt;td&gt;E08&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;F01&lt;/td&gt; &lt;td&gt;RESPECTIVE_BILLER_GL&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to GL&lt;/td&gt; &lt;td&gt;CL_CHQ_REALiZE_TO_GL&lt;/td&gt; &lt;td&gt;E09&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E78&lt;/td&gt; &lt;td&gt;CLEARING_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Dishonour of cheque before value date&lt;/td&gt; &lt;td&gt;CL_CHQ_DISHONOR_PRE_VALUEDATE&lt;/td&gt; &lt;td&gt;E10&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E11&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Dishonour of cheque after value date&lt;/td&gt; &lt;td&gt;CL_CHQ_DISHONOR_POST_VALUEDATE&lt;/td&gt; &lt;td&gt;C42&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E12&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Reject of O/w </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Clearing</td> <td>CL_OUTWARD_CHQ_REJECT</td> <td>E13</td> <td>N</td> <td>CLEARING_ENDPOINT</td> <td>CHEQUE_AMOUNT</td> <td>E14</td> <td>OUTWARD_REJECT_SUSPENSE</td> <td>CHEQUE_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> <tr> <td>Outward Clearing Reject to Pass</td> <td>CL_OUTCHQ_REJECT_TO_PASS</td> <td>E15</td> <td>N</td> <td>OUTWARD_REJECT_SUSPENSE</td> <td>CHEQUE_AMOUNT</td> <td>E16</td> <td>CLEARING_CFC</td> <td>CHEQUE_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> <tr> <td>Return of partial immediate credit</td> <td>CL_RETURN_CHQ_PART_IMM_CREDIT</td> <td>C43</td> <td>N</td> <td>CLEARING_CASAACCOUNT</td> <td>IMMEDIATE_CREDIT_AMOUNT</td> <td>E17</td> <td>CLEARING_CFC</td> <td>IMMEDIATE_CREDIT_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> <tr> <td>Return of partial immediate credit</td> <td>CL_RETURN_CHQ_PART_IMM_CREDIT</td> <td>E18</td> <td>N</td> <td>CLEARING_CFC</td> <td>CHEQUE_AMOUNT</td> <td>E19</td> <td>CLEARING_ENDPOINT</td> <td>CHEQUE_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> </table> @param transactionLog @param accountingEventAmountTagDetailsDTOArray @throws FatalException @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	{@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense {@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)} @param transactionContext @param transactionLog
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method raise prepared the credit leg and raise the accounting by calling accounting module.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method raises accounting while standing instruction batch execution and the biller is consolidated. @param applicationContext @param accountingEventAmountTagDetailsDTOs @param transactionAmount @param narration @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will be used if during biller registration the biller unchecks the consolidated remittance option. @param applicationContext @param billTransaction @param listAccountingEventAmountTagDetailsDTO @param eventId @param isConsolidated @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method is used to raise accounting event on successful bill payment transaction. @param applicationContext @param eventBillPay @param remitComponentPreferenceItemDTO @param listAccountingEventAmountTagDetailsDTO @param transactionAmount @param transactionDescription @param isCallee @param isNotConsolidated @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<p>This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response.</p> <pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to CASA&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_CASA&lt;/td&gt; &lt;td&gt;E38&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C52&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to Loan&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_LOAN&lt;/td&gt; &lt;td&gt;E39&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;L02&lt;/td&gt; &lt;td&gt;RPA&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to TD&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_TD&lt;/td&gt; &lt;td&gt;E40&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;T65&lt;/td&gt; &lt;td&gt;CLEARING_BRIDGE_GL&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to CASA Reject&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_CASA_REJ&lt;/td&gt; &lt;td&gt;E41&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E42&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to Loan Reject&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_LOAN_REJ&lt;/td&gt; &lt;td&gt;E43&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E44&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to TD Reject&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_TD_REJ&lt;/td&gt; &lt;td&gt;E45&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E46&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to CASA Return&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_CASA_RETURN&lt;/td&gt; &lt;td&gt;E47&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E48&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to Loan Return&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_LOAN_RETURN&lt;/td&gt; &gt; &lt;td&gt;E49&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E50&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to TD&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_TD_RETURN&lt;/td&gt; &lt;td&gt;E51&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E52&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to CASA-debit to heldover&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_CASA_HELDOVR&lt;/t d&gt; &lt;td&gt;E53&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;C53&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to TD-debit to heldover&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_TD_HELDOVR&lt;/td&gt; &lt;td&gt;E54&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;L03&lt;/td&gt; &lt;td&gt;CLEARING_BRIDGE_GL&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to Loan-debit to heldover&lt;/td&gt; &lt;td&gt;PY_INW_PAYMENT_TO_LOAN_HELDOVR&lt;/td&gt; &lt;td&gt;E55&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;T66&lt;/td&gt; &lt;td&gt;RPA&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Endpoint to heldover movement&lt;/td&gt; &lt;td&gt;PY_MOVEMENT_ENDPOINT_HELDOVR&lt;/td&gt; &lt;td&gt;E56&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E57&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to CASA Reject to Pass&lt;/td&gt; &lt;td&gt;PY_INW_PMT_CASA_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;E79&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C73&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to TD Reject to Pass&lt;/td&gt; &lt;td&gt;PY_INW_PMT_TD_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;E80&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;T68&lt;/td&gt; &lt;td&gt;CLEARING_BRIDGE_GL&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Payment to Loan Reject to Pass&lt;/td&gt; &lt;td&gt;PY_INW_PMT_LOAN_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;E81&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_UNPOSTED_CREDIT&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;L04&lt;/td&gt; &lt;td&gt;RPA&lt;/td&gt; &lt;td&gt;INWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param transactionLog @param accountingEventAmountTagDetailsDTO @throws FatalException </pre>
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	for more info {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	for more info {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will accept the credit leg from settlement execution and raise accounting event for tc encashment.   @fcb.param ApplicationContext,applicationContext @fcb.param AccountingEventAmountTagDetailsDTO[],amountTagDetailsDTO @fcb.param String,eventId @fcb.param MoneyDTO,payOutAmount @fcb.param String,remarks @fcb.param SettlementModeType,settlementModeType @throws FatalException @see ApplicationContext @see AccountingEventAmountTagDetailsDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Debit&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_DEBIT&lt;/td&gt; &lt;td&gt;C44&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E20&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT&lt;/td&gt; &lt;td&gt;E21&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E22&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Pass to Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_PASS_TO_REJECT&lt;/td&gt; &lt;td&gt;E23&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C45&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject to Pass&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;C46&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E24&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;O/w Clearing for I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_OUTWARD_INWARD_REJECTS&lt;/td&gt; &lt;td&gt;E25&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E26&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Float Extension&lt;/td&gt; &lt;td&gt;CL_CHQ_FLOAT_EXTENSION&lt;/td&gt; &lt;td&gt;E27&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E28&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; </pre>
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {AccountingEventAmountTagDetailsDTO} based on the account debit response. <style type="text/css">table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;}</style>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre>.tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;&amp;nbsp;&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PRECUTOFF&lt;/td&gt; &lt;td&gt;C54&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;E58&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PSTCUTOFF&lt;/td&gt; &lt;td&gt;C55&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E59&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PRECUTOFF&lt;/td&gt; &lt;td&gt;E60&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;E61&lt;/td&gt;</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PSTCUTOFF&lt;/td&gt; &lt;td&gt;E62&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E63&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_MOVEMENT_HELDOVR_ENDPOINT&lt;/td&gt; &lt;td&gt;E64&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E65&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_PRECUT&lt;/td&gt; &lt;td&gt;E66&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;C56&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_POSTCUT&lt;/td&gt; &lt;td&gt;E67&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C57&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from GL&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_GL&lt;/td&gt; &lt;td&gt;E68&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E69&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_RECALL_MVMT_HELDOVR_ENDPT&lt;/td&gt; &lt;td&gt;E70&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E71&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment for ASP Verification&lt;/td&gt; &lt;td&gt;PY_OUTPAY_ASP_VERIFY&lt;/td&gt; &lt;td&gt;E75&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_MISC_EXP&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E76&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param transactionLog @param amountTagDetailsDTO @param messageLogItem @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<p>This method handles accounting for transactions where a fraud check is required. &lt;br&gt; &lt;br&gt; If the transaction source is settlement, accounting is not be raised by payments. It will be done by the caller module. Payments will only build a basic derived intermediary gl credit leg and send it out to the caller. &lt;br&gt; &lt;br&gt; If the transaction source is not settlement,&lt;br&gt; If the account involved is CASA, debit CASA and credit intermediary gl will be raised here. Debit CASA leg will be coming in the input which will be updated here for the amount tag id and since intermediary gl is derived, only a basic credit leg will be built. Event raised will be &lt;br&gt; If the account involved is GL, debit GL and credit intermediary gl will be raised here. Debit GL leg will be coming in the input which will be updated here for the amount tag id. The credit leg will not be built in this case as both the amount tags are the same. Event raised will be PY_OUTPAY_DEBIT_GL_PRECUTOFF, if transaction is pre-cutoff or PY_OUTPAY_DEBIT_GL_PSTCUTOFF, if transaction is post-cutoff. &lt;/p&gt;           @param transactionContext            @param transactionLog @param amountTagDetailsDTO @param messageLogItem            @return @throws FatalException         </p>
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	For accounting {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	for more info {@link OutgoingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)} @param transactionContext @param transactionLog @param amountTagDetailsDTO @param eventId @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method builds the accounting leg and triggers accounting for Business Payments. </br> @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param messageLogItem @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method does the reject accounting for manager cheque. @param accountingDetail @param txnContext @param newTransactionLog @param messageLogItem TODO @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	Method to do debit accounting for consolidated bill payment @param accountingEventAmountTagDetailsDTOs @param transactionAmount @param narration @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	@param transactionContext @param transferTransactionLog @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This service will call prepareAccountingLeg which will be used to override the accounting tag details which is received after settlement with MC accounting tags.  <li>It will also raise the accounting for the bank draft issuance. <li>If bank draft is issue via settlement then build the settlement bridge GL leg @param accountingLegs . Accounting tag details. @param IsSettlementSource @param entity entity has all the information related to bankDraft (Settlement mode, Amount, serialnumber, etc)
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will raise an accounting for the bank draft liquidation only. @param transactionContext @param bankDraft @param eventId @param accountingEventAmountTagDetailsDTO @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This service will do the unclaimed accounting. <li>This service will move the funds from the bank draft GL to the unclaimed GL. @param transactionContext @param bankDraft @return
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	It undoes the transaction and raise dishonor accounting. @param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do the re-process accounting of refused transactions. @param transactionContext @param transactionLog @param messageLogItem @param reverse @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	@param transactionContext @param transactionLog @param messageLogItem @param accountingEventAmountTagDetailsDTO @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do the refusal accounting of the dishonor of outward transactions @param transactionContext @param transactionLog @param messageLogItem @param reverse TODO @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method will do the accounting for CASA debit process and GL credit.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO[], amountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	This method provides the common suspense handling mechanism. A suspense can happen, when the transaction must honored inspite of failure. This will happen if the transaction is posted as advice (force post) @param transactionContext TODO @param messageItem TODO @param txnLog @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Collection of CASA posting&lt;/td&gt; &lt;td&gt;CT_INW_COLL_CASA_POSTING&lt;/td&gt; &lt;td&gt;C50&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_ACCOUNT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;E35&lt;/td&gt; &lt;td&gt;COLLECTIONS_ENDPOINT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;tr&gt; &lt;td&gt;Inward Collection of CASA reject&lt;/td&gt; &lt;td&gt;CT_INW_COLL_CASA_REJECT&lt;/td&gt; &lt;td&gt;P02&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_UNPOSTED_DEBIT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;E36&lt;/td&gt; &lt;td&gt;COLLECTIONS_ENDPOINT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT_EQVLNT&lt; /td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Collection of CASA return&lt;/td&gt; &lt;td&gt;CT_INW_COLL_CASA_RETURN&lt;/td&gt; &lt;td&gt;E37&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_ENDPOINT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;P03&lt;/td&gt; &lt;td&gt;COLLECTIONS_UNPOSTED_DEBIT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT_EQVLNT&lt; /td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Inward Collection of CASA override&lt;/td&gt; &lt;td&gt;CT_INW_COLL_CASA_OVERRIDE&lt;/td&gt; &lt;td&gt;C51&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_ACCOUNT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;P04&lt;/td&gt; &lt;td&gt;COLLECTIONS_UNPOSTED_DEBIT&lt;/td&gt; &lt;td&gt;INWARD_COLLECTION_AMOUNT_EQVLNT&lt; /td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; </pre>
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	<pre> @see IncomingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem) com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc. </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				entity .log.TransactionLog)
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	{@link PersonalChequeService#doInwardAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[])} @see com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.entity .log.TransactionLog)
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @param accountingRequest @throws FatalException	{@link PersonalChequeService#doInwardAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[])}
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	This method will reverse the accounting entries for a transaction processed earlier. @param pcReversalAccountingEventDTO @exception throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	This method reverse the accounting that has been raised while original transaction. @param sessionContext @param originalReferenceNo @param eventId @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	This method is used to do the reversal accounting @param sessionContext @param bankDraftInstrument @param eventId @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	This method is used to do the reversal accounting @param transactionContext @param bankDraftInstrument @param eventId @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	This method will do the refusal accounting of the dishonor of outward transactions @param transactionContext @param transactionLog @param messageLogItem @param reverse TODO @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	raiseReversalAccountingEvent	This method raise reverse accounting event after successful account transaction. @param pcReversalAccountingEventDTO @throws FatalException	@param applicationContext @param transactionLog @param messageLogItem @param reversalEventCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	populateAndRaiseAccountingEventForBillPayment	This method will raise accounting event after successful bill pay transaction. @param transactionReferenceNo @param eventBillPay @param remitComponentPreferenceItemDTO @param listAccountingEventAmountTagDetailsDTO @param transactionAmount @param transactionDescription @param isCallee @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	This service fetches the cheque status details @param sessionContext @param chequeStatusInquiryDTO
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	This method is validating the ledger code. @param sessionContext @param ledgerCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	@param log @param taskModuleType @param internalAccountDetailDTO @param internalAccountDetail @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	This method will fetch very basic details for the supplied customer account and if details are not found it will throw an error. @param instrumentTransferTransactionLog @param internalAccountDetailDTO @param internalAccountDetail @return @throws FatalException
IPCAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCAccountingAdapter	validateLedgerCode	<p>This method validates the input ledger code when a fund exchange transaction is initiated for a ledger.</p> @param ledgerCode @return @throws FatalException	
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	approveTask	This method approves task for input workflow id @param workflowId @throws FatalException	This method approves task for input workflow id @param workflowId @throws FatalException
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	approveTask	This method approves task for input workflow id @param workflowId @throws FatalException	This method approves task for input workflow id and reason codes. @param workflowId, String @param reasonCodes, List<String> @param comment, String @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	This method fetches workitem for the provided keyAsString value and then checks whether the operation represented by the provided maintenanceType can be performed on the current transaction. @param sessionContext @param keyAsString @param maintenanceType @param userId @throws FatalException
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	This method retrieves assigned tasks for input user @param userId @return @throws FatalException
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireTaskDetails	This method retrieves task details for input workflowId @param workflowId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	getAssignedTasksForUser	This method returns assigned tasks for input user @param userId @return @throws FatalException	This method retrieves assigned tasks for input user @param userId @return @throws FatalException
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	getAssignedTasksForUser	This method returns assigned tasks for input user @param userId @return @throws FatalException	
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireAllPendingTasks	@return @throws FatalException	
IWorklistTaskQueryServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistTaskQueryServiceAdapter	inquireAllPendingTasks	@return @throws FatalException	
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchPartyInquiryResponse	This method fetches PartyInquiryResponse, which is then used for validating credentials of the user. @param sessionContext @param partyId @return {@link PartyInquiryResponse} @throws FatalException	This method returns the instance of channel adapter. @return ChannelPartyAdapter class instance

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchPartyInquiryResponse	This method fetches PartyInquiryResponse, which is then used for validating credentials of the user. @param sessionContext @param partyId @return {@link PartyInquiryResponse} @throws FatalException	This method validates parameters provided by the user with those in the database. The method first fetches the emailID using the partyID from the database and validates it. It then fetches the partyID from accountID and validates the same with the accountID mentioned in the mail body. @param sessionContext the session context to be used. @param emailTransactionContext contains all the information required to execute the email transaction. @throws FatalException
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchPartyInquiryResponse	This method fetches PartyInquiryResponse, which is then used for validating credentials of the user. @param sessionContext @param partyId @return {@link PartyInquiryResponse} @throws FatalException	This method fetches {@link PartyInquiryResponse}. @param sessionContext @param partyId @return PartyInquiryResponse @throws FatalException
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchAllPartiesOfAccount	This method fetches {@link AccountHolderInquiryResponse}, which is then used for validating credentials of the user.. @param accountID @return AccountHolderInquiryResponse @throws FatalException	This method returns the instance of channel adapter. @return ChannelPartyAdapter class instance
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchAllPartiesOfAccount	This method fetches {@link AccountHolderInquiryResponse}, which is then used for validating credentials of the user.. @param accountID @return AccountHolderInquiryResponse @throws FatalException	This method validates parameters provided by the user with those in the database. The method first fetches the emailID using the partyID from the database and validates it. It then fetches the partyID from accountID and validates the same with the accountID mentioned in the mail body. @param sessionContext the session context to be used. @param emailTransactionContext contains all the information required to execute the email transaction. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPartyAdapter	fetchAllPartiesOfAccount	This method fetches {@link AccountHolderInquiryResponse}, which is then used for validating credentials of the user.. @param accountID @return AccountHolderInquiryResponse @throws FatalException	This method fetches {@link PartyAccountRelationshipInquiryResponse} @param accountID @return PartyAccountRelationshipInquiryResponse @throws FatalException
IPartyRelationshipAccessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyRelationshipAccessAdapter	fetchLegalName		
IProductManufacturingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IProductManufacturingAdapter	fetchAllProductGroupDetails	@param applicationContext @param productGroupCode @param businessUnit @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	updateAccountRepository	This method is called after successful account opening from origination. @param sessionContext @param accountId @return TransactionStatus	This method is used update the Account Configurations for a Demand Deposit Account that has already been opened previously. Account level preferences such as statement preferences, minimum balance, debit card preferences, account facilities such as sweep in sweep put instructions, etc. are updated. Additionally, operating instructions both financial and non financial for the account, cheque book requests and default communication address for the account are also updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration. This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	updateAccountRepository	This method is called after successful account opening from origination. @param sessionContext @param accountId @return TransactionStatus	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchDefaultAccountAddress	This method is used to inquire the default communication address for a DDA account. @param sessionContext @param accountId @throws FatalException	This method gets the address of account for deliver/dispatch/request of cheque book @param accountId @param chequeBookRequest @return
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	linkDDAccountToOtherBranch	This method is used to link the DDAAccountBranchTransferApplicationService for CASA Account Branch Transfer @param sessionContext @param ddAccountBranchTransferDetailsDTO @throws FatalException	Method linkModuleBranchTransfer is being used to decide the service call based on moduleType. i.e. if account type is CASA call method linkDDAccountToOtherBranch(sessionContext, ddAccountBranchTransferDetailsDTO) Method to link Module wise Account Branch Transfer.
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	collectLegalOrOutFee		
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	writeOffFeeInHardshipRelief	This method will be called from FeesManagementService to write Off the fee while account is in hardship relief period. @param accountId @param eventId @param nsfTreatment @param feeAmount @param arrearType @param feeCode	This method will be called from FeesManagementService to write Off the fee while account is in hardship relief period. @param accountId @param eventId @param nsfTreatment @param feeAmount @param arrearType @param feeCode
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	writeOffFeeInHardshipRelief	This method will be called from FeesManagementService to write Off the fee while account is in hardship relief period. @param accountId @param eventId @param nsfTreatment @param feeAmount @param arrearType @param feeCode	This method will be called to write off the charged fee during hardship relief period. This method will call write-off service from DDAccountAdapter for writing off fee. @param feeAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	writeOffFeesInHardshipRelief	This method will be called from FeesManagementService to write Off the fee while account is in hardship relief period. @param accountId @param eventId @param nsfTreatment @param feeAmount @param arrearType @param feeCode	This method will be called to write off the charged fee during hardship relief period. This method will call write-off service from DDAccountAdapter for writing off fee. @param feeAmount
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	raiseArrearInHardshipRelief	This method will be called from FeesManagementService to raise an Arrear for the fee charged amount during Hardship relief period. @param accountId @param postArrearRunningBalance @param eventId @param priceDefinitionId	This method will be called from FeesManagementService to raise an Arrear for the fee charged amount during Hardship relief period. @param accountId @param postArrearRunningBalance @param eventId @param priceDefinitionId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	isSavingAccount	This method return true if the account is saving. @param accountId	This method return true if the account is saving. @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	isSavingAccount	This method return true if the account is saving. @param accountId	This method validates if Account is checking or saving. If the account is saving it will raise a business exception. @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchTotalOffsetBenefitAmount	This method will be used to inquire total offset benefit on account. @param accountId @return BenefitInquiryResponse	This method will be used to inquire total offset benefit on account. @param accountId @return BenefitInquiryResponse
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchTotalOffsetBenefitAmount	This method will be used to inquire total offset benefit on account. @param accountId @return BenefitInquiryResponse	Method will populate details of BalanceInquiryDetailsDTO using entity {@link IDemandDepositAccount}; @param demandDepositAccount @return BalanceInquiryDetailsDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	isFacilityLinked	This method returns true if the account is over draft and facility linked to it. @param accountId	This method validates if Account is checking or saving. If the account is saving it will raise a business exception. @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	validateAccountSystemStatus	This method returns AgingAccountStatus. @param accountId	This method validates if Account is checking or saving. If the account is saving it will raise a business exception. @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	isAccountBlocked	This method returns true if account is blocked. @param accountId	This method returns true if account is blocked. @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchODFacilityExpiryDate		This method will be call in case of change term relief. @param accountId @param extendedExpiryDate
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchODFacilityExpiryDate		This method will be call in case of change term relief. @param accountId @param extendedExpiryDate
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountAddressLinkage	This method creates the account address linkage in case of account opening through origination @param sessionContext @param accountAddressLinkageDTO	This method creates the account address linkage in case of account opening through origination @param sessionContext @param accountAddressLinkageDTO
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountAddressLinkage	This method creates the account address linkage in case of account opening through origination @param sessionContext @param accountAddressLinkageDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountAddressLinkage	This method creates the account address linkage in case of account opening through origination @param sessionContext @param accountAddressLinkageDTO	This method would create the account address linkage while creating a account @param accountId @param partyId @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllOffsetAccount	This service will return all the principal offset account for supplied beneficiary account. @param accountId @return List<String>	This service will return all the principal offset account for supplied beneficiary account. @param accountId @return List<String>
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllOffsetAccount	This service will return all the principal offset account for supplied beneficiary account. @param accountId @return List<String>	<p> This checks if the new offer ID is linked to offset offer </p> @param ddOfferSwapBusinessPolicyDTO @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchEligibleOffersForLinkageSet	This service will return the set of all eligible offset offer. @param accountId @param currencyCode @return Set<String>	<p> This checks if the new offer ID is linked to offset offer </p> @param ddOfferSwapBusinessPolicyDTO @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchOperatingInstructions	This method would fetch the operating instructions for particular operating instruction DTO	This method is used to populate the response message in case of successful transaction, for that it is using the input requestDTO and response for the each case. Inside the method it checks the type of transaction and based on that it populates use case based outputs. @param sessionContext @param fundTransferRequestDTO @param fundTransferResponse @param identifierType @param transactionAmount @param initiatorAccountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllProviderAccountsOfSweepInSetup	This method is used to fetch all the provider accounts for given beneficiary sweep in account @param accountId @return	This method is used to fetch all the provider accounts for given beneficiary sweep in account @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllProviderAccountsOfSweepInSetup	This method is used to fetch all the provider accounts for given beneficiary sweep in account @param accountId @return	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountMemo	This method is used to create account level memo from demand deposit account	This method is used to create account level memo from demand deposit account
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountMemo	This method is used to create account level memo from demand deposit account	This method would add the warning indicator audit item for given account. @param sessionContext @param warningIndicatorDTO @param accountId @param primaryReasonCode
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	createAccountMemo	This method is used to create account level memo from demand deposit account	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will either add or update or remove a warning indicator for a given account id when a new warning indicator is added or updated at party level @param warningIndicators @param accountId @param moduleType @throws FatalException @throws OverridableException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will publish the warning indicators to the account @param sessionContext @param warningIndicatorToKeyMap @param removedWarningIndicatorToKeyMap @param warningIndicatorAuditItemList @param warningIndicatorList @param warningIndicatorDetailDTO @param warningIndicatorInquiryResponse @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException @throws OverridableException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will prepare the warning indicator map to be used in intimation of collections @param warningIndicatorToKeyMap @param removedwarningIndicatorToKeyMap @param warningIndicatorAuditItemList @param warningIndicatorList @param warningIndicatorDetailDTO @param warningIndicatorDetailsKeyDTO
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method would remove the recovery warning indicator on account when called from collections module @param sessionContext @param recoveryAccountDetailsDTO @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method would check the recovery warning indicator and remove the same from account level @param sessionContext @param recoveryAccountDetailsDTO @param warningIndicatorSummaryService @param acctWIActionSummaryDtlDTO @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will either add or update or remove a warning indicator for a given account id when a new warning indicator is added or updated at party level @param warningIndicators @param accountId @param moduleType @throws FatalException @throws OverridableException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will be used through the impacts screen for addition/removal of warning indicators. @param sessionContext @param accountWIDetailDTO @param warningIndicatorDTO @throws FatalException @throws OverridableException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method is used when warning indicator is modified through account warning indicator impacts screen @param sessionContext @param accountWIDetailDTO @param warningIndicatorInquiryResponse @param warningIndicatorDetails @param i @param warningIndicatorDetailsKeyDTO @throws FatalException @throws OverridableException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method is used for fetching the warning indicators against an given account id
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method is used for creating the warning indicators map against an given account id
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method would prepare map for warning indicator action based on warning indicator set on account level @param accountWIActionSummaryMap @param accountWIActionSummaryDtlDTOs @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method will decide if the account with warning indicator has over-ride status as Yes @param accountWIActionSummaryDtlDTOs @param isOverridenStatus @param accountWIActionSummaryDtlDTO @param tempMap @return @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method would fetch all the warning indicators, warning indicator details from the x-ref @param reasonCode @param casaWarningIndicatorType @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method check if there are any joint parties for given account with warning indicators @param warningIndicatorTypeForRemoval @param noWarningIndicatorOnAccount @param sessionContext @param dDAPartyAdapter @param partyAccountRelationshipDTO @return
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param reasonCode @param casaWarningIndicatorType
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param newDebitBlockReasonCodeSet @param newTotalBlockReasonCodeSet @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param totalBlockReasonCode @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param totalBlockReasonCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param totalBlockReasonCode @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param totalBlockReasonCode @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchWarningIndicator	This method is used to fetch all the Warning indicators set in the warning indicator x-ref @param warningIndicatorDTO @return @throws FatalException	@param accountId @param debitBlockReasonCode @param warningIndicator @throws FatalException
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	removeAccountMemoForWarningIndicator	This method is used to remove account level memo from demand deposit account	This method will remove the warning indicator audit item for a given account. @param warningIndicatorDTO @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllAccountWarningIndicators	This method will fetch all the warning indicators set at the account.	This represents the fetchWarningIndicator method of the entity. This method will fetch all the account level warning indicators and their corresponding actions set on the account. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,WarningIndicatorDTO,warningIndicatorDTO, ,warningIndicatorDTO TODO: Explain the warningIndicatorDTO above @return {@link com.ofss.fc.app.account.dto.warningindicator.WarningIndicatorResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllAccountWarningIndicators	This method will fetch all the warning indicators set at the account.	This method would add/delete multiple warning indicators for a particular account, the account level warning indicators can be added for a particular account or the actions can be over-riden on account level
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAllAccountWarningIndicators	This method will fetch all the warning indicators set at the account.	This method will fetch all the warning indicators set at the account.
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAccountMemo	This method is used to fetch account level memo from demand deposit account	This method is used to fetch account level memo from demand deposit account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAccountMemo	This method is used to fetch account level memo from demand deposit account	This method would add the warning indicator audit item for given account. @param sessionId @param warningIndicatorDTO @param accountId @param primaryReasonCode
IDDAAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAAccountAdapter	fetchAccountMemo	This method is used to fetch account level memo from demand deposit account	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre> @param sessionId @param productId @param currencyCode TODO @return	This method fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> @param applicationContext @param accountId @param eventToComputeInterest @param loanAccount </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&amp;lt;br&amp;gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmoutrnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre> <p>@param sessionId @param productId @param currencyCode TODO @return</p>	<pre> The method will be used to inquire Partial pay off details. The method will validate if partial pay off is allowed for the account and the account is a valid, disbursed account. The method will return minimum transaction amount to perform PPF and account balances. Minimum partial pay off amount calculation on the basis of minimum number of installments for partial pay off maintained at product level. </pre> <p>MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. The parameter represents valid loan account for which partial payoff amount is to be inquired.</p> <p>MI,boolean,isUnbilledInterestToBeCharged, ,This flag tells system to whether Unbilled Interest will be charged as a part of partial pay off transaction.</p> <p>com.ofss.fc.app.loan.dto.LoanPartialPayOffInquiryResponse} throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for Transaction Failure section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre>&lt;pre&gt;This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc &lt;/pre&gt; @param sessionContext @param productId @param currencyCode TODO @return</pre>	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre>&lt;pre&gt;This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc &lt;/pre&gt; @param sessionContext @param productId @param currencyCode TODO @return</pre>	<p>This method does partial payoff for a loan account through the provided CASA account. It provides partial payoff functionality through a simple API and as required by Dashboard. @param sessionContext @param CASAAccountNo @param loanAccountNo @param chequeNo @param chequeDate @param ppfAmount @return LoanPartialPayOffInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> This method does partial payoff for a loan account through Cash. It provides partial payoff functionality through a simple API and as required by Dashboard. @param sessionContext @param loanAccountNo @param ppfAmount @param currencyDenominationDTO @return LoanPartialPayOffInquiryResponse @throws FatalException </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> This method does partial payoff for a loan account through RPA. It provides partial payoff functionality through a simple API and as required by Dashboard. @param sessionContext @param loanAccountNo @param ppfAmount @return TransactionStatus @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> This method provides facility to pay Loan partially by External Account.&lt;br&gt; It has been exclusively defined for DashBoard Application. @param sessionContext @param externalAccountNumber @param loanAccountNumber @param ppfAmount @return @throws FatalException </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> Validate that the remaining term of the parent loan account should be greater than or equal to the "Minimum term for debt consolidation" defined in the Loan Product Preference. Loan consolidation will only be allowed if the remaining loan term of the parent loan account is greater than or equal to the "Minimum term for debt consolidation" as of the date of consolidation. </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre>	<pre> 1)This method calls fetchLoanProduct of loanProductApplication .fectchLoanProduct returns loanProduct entity. 2)This entity passed to method "populateLaonProductPreferencesDTO" which returns LoanProductPreferencesDTO. </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductPreferences	<pre> This method is responsible for fetching the Loan Product Preferences from the Product Manufacturing Module Loan Product Preferences contain the attributes that are defined by a product manager in a bank . They are parameters which do not change much . Some of the parameters are :- 1)Appropriation Sequence 2)Minimum Loan Term 3)Maximum Loan Term 4)Minimum Loan Amount 5)Maximum Loan Amount etc </pre> @param sessionContext @param productId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	This method is a common method for populating {@link LoanRepaymentHoliday} in to {@link LoanRepaymentHolidayBusinessPolicyDTO} and invoking {@link LoanRepaymentHolidayBusinessPolicy}. @fcb.param applicationContext @fcb.param loanRepaymentHoliday

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	This method is called to update the limit utilized by a Loan Account.  Whenever Arrears are raised or When installment amount is paid the customer LCM needs to be intimated to update the limit utilized.  For all the Arrears raised or appropriated LCM needn't be intimated. Arrears for which LCM needs to be intimated in configurable at OfferLevel.  LCM is intimated with the pre and post balance snapshot of the Arrear for which limit is utilized. This method works as follows :=   <ul> <li>Fetch the Loan Account details using {@link LoanAccountService#fetchLoanAccount(accountId)}</li> <li>Fetch the Offer details using {@link LoanProductAdapter.fetchLoanProductOfferDetails(null,loanAccount.getProductId(), loanAccount.getOfferId())}</li> <li>Check for each ArrearType the limit has been configured or not. If the limit is configured then intimate LCM using {@link UtilizationApplicationService#updateAccountUtilization()}.</li> </ul>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	This method fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	<pre>This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&lt;br&gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmountnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre> @param productId @param offerId @param currencyCode TODO @return product offerId @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	@param accountId @param startDate @param noOfHolidaysRequested
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	<pre> 1)This method calls fetchLoanProduct of loanProductApplication .fectchLoanProduct returns loanProduct entity. 2)This entity passed to method &quot;populateLaonProductPreferencesDTO&quot; which returns LoanProductPreferencesDTO. </pre> @param sessionContext @param productId @return LoanProductPreferencesDTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanProductOfferDetails	This method is responsible for returning a consolidated response containing the product preferences and the offer details. @param context @param productId @param offerId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	<p>This method fetches the Offer details for the provided Loan Account Id. </p> @param sessionContext @param loanAccount @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method checks offer details for loan account. It checks loan Account have EFS as allowed transaction type. It checks loan account is in lockin period or not. If loan account is in locking period it raises an error. @param loanAccount
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method validates the Accelerated Payment instructions present @param sessionContext @param loanAccountAmendConfigurationDTO @param amendTypes

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	<p>This method fetches the Offer details for the provided Loan Account Id. </p> @param sessionContext @param loanAccount @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This methods fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	@param accountId @param offerId @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param	This method fetches the LoanAccount Details of a Loan account to be amended @param sessionContext @param loanAccountAmendConfigurationDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	<pre>This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&lt;br&gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmountnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre> @param productId @param offerId @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				currencyCode TODO @return product offerId @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	<p>This method fetches the Offer details for the provided Loan Account Id. </p> @param sessionContext @param loanAccount @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method is responsible for performing Loan Offer Swapping @param sessionContext @param offerSwapDetailsDTO
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method updates the Offer and schedule related attributes. @param sessionContext @param offerSwapDetailsDTO @param currencyCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	Validate that the consolidated amount for the parent account does not exceed the maximum loan amount defined for Loan account's offer
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method populates the common account feeder information to feederDTO. @param loanAccount @param feederDTO @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param	<pre> 1)This method calls fetchLoanProduct of loanProductApplication .fetchLoanProduct returns loanProduct entity. 2)This entity passed to method &quot;populateLaonProductPreferencesDTO&quot; which returns LoanProductPreferencesDTO. </pre> @param sessionContext @param productId @return LoanProductPreferencesDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param sessionContext @param productCode @param branchCode @param currency
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method is used to initiate penalty interest computation from due date. First it checks for unpaid arrears. If there are no unpaid arrears, it performs a regular interest computation otherwise it performs penalty computation from due date. @param applicationContext @param effectivedate @param toleranceAmount @param loanAccount
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method checks whether penalty tolerance is breached or not if so then only compute penalty. @param accountId @param effectivedate @param toleranceAmount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method will generate a Payout Quote for an existing loan of the customer on the basis of loan Event Type. Quotes can be maintained for Partial Payoff(PPF)/Final Settlement(EFS)/Rate Change(Loan Event Type). A unique quote ID will be generated for each Quote generation. @param sessionContext @param loanEventSimulationDTO @return LoanSaveEarlyPayoutQuoteResponse @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param	This method populates {@link LoanEventSimulationBusinessPolicyDTO}. @param loanEventSimulationDTO @return loanEventSimulationBusinessPolicyDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			currencyCode TODO @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	This method will return LoanServeInquiryResponse for loanAccount .This is required for Loan Serve Make Payment Transaction. @param sessionContext @param accountId @return LoanServeInquiryResponse @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanOfferDTO	This method is responsible for fetching the Loan Offer Details Offers are maintained in Product Manufacturing module . Attributes / Parameters / Functionalities that can be marketed are setup in the Offer @param sessionContext @param offerId @param currencyCode TODO @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateProduct	This method is responsible for checking the product. @param context @param productId @param offerId @return @	<pre> This method creates a new loan account. It performs the following functions:- &lt;li&gt;validating the product Id. &lt;/li&gt;validate the offer ID. &lt;/li&gt;ELCM call to validate the lines. &lt;/li&gt;Validates the input details as well as data from CIF modules. &lt;/li&gt;Generates the loan Account Number as per the configuration maintained. &lt;/li&gt;Open the loan Account. &lt;/li&gt;Updating Loan Account Addition Details. &lt;/li&gt;Logs activity of Alert processing. &lt;/li&gt;fcb.param MI,ApplicationContext,applicationContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. &lt;/li&gt;fcb.param MI,LoanAccountOpeningDTO,loanAccountOpeningDTO. This holds the details required when opening a loans account. The transaction status will indicate whether the transaction went through successfully and causes the error if unsuccessful Success &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_SUCCESS}&lt;/li&gt; &lt;/ul&gt; Transaction failure Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EBA_PROD_NOTMAPPED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EGL_TXN_NO_INV}&lt;/li&gt; &lt;/ul&gt; @return {@link com.ofss.fc.app.loan.dto.account.LoanAccountResponse} This response object holds information of the newly generated account and the status of transaction. @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateProduct	This method is responsible for checking the product. @param context @param productId @param offerId @return @	<pre>This method is used to check whether the product is valid or not by inquiring the product module It checks whether the given product is in active state at the product module side or not, In case if it active it returns true else returns false </pre> @param sessionContext @param productCode @return boolean @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateProduct	This method is responsible for checking the product. @param context @param productId @param offerId @return @	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method is used to validate UDI. @param sessionContext @param accountId @param repaymentRecordsDTO @param repaymentStageDTO @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This methods fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method first fetches the group which the given product code belongs to . then fetches the corresponding group's preferences

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	<p>&lt;p&gt;This method is used to perform validations with respect to facility Details. &lt;br&gt; It performs following validations during Loan Account Opening:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To make sure whether facility is available or not.&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Account creation date resides in between Facility Period Dates&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount&lt;/li&gt;</li> </ul> <p>&lt;br&gt; If above tests are successful, then it performs below mentioned tasks:-</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To link Loan Account to Facility.&lt;/li&gt;</li> <li>&lt;li&gt;To raise an Earmark for the Loan Account.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @param applicationContext @param loanAccountId @throws FatalException</p>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	<p>This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	<pre>This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&lt;br&gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmountnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre> @param productId @param offerId @param currencyCode TODO @return product offerId @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method first fetches the group which the given product code belongs to . then fetches the corresponding group's preferences
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method fetches rate lock information. 1> fetches product group code. 2>fetches credit policy template id from product group code (calling loan product adapter method) fetches the credit template details on the basis of credit policy Template .Sets the value in LoanProductPreferenxeDto and business unit.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param offerId @param pricingTicketNumber @param productId @param fixedTermInMonths @param newApprovedAmount @param newLoanTerm @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method prepares {@link DocumentDTO} which is required for report operation. This contains the documentTypeId used from {@link LoanStatementPrintConfiguration} file. It also contains the productGroup. @param demandDepositAccount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	<pre>This method is called to fetch fetchMainLine's productCode. The following steps are carried out in this method:- &lt;br&gt; 1)MainLine Id is fetched by calling fetchMainFacilityForAccount method of LimitsAccountApplicationService(). 2)MainLine details are fetched by calling method LoanLCMAAdapter.fetchFacilityDetails(String). 3)Then mainline's productGroupCode is fetched from the response. 4)Call is given to fetchProductGroupDetails(ApplicationContext, String) to check whether the mainline's productgroup is MOF or NOT. 5)If mainline's product group is MOF then product group code of mainline is returned. </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method fetches productGroup preferences for the given product group code.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method Credit Policy Code on basis on business unit, and product group code.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method will fetch all the details of the product group. @param sessionContext @param productGroupCode
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroup	This method is responsible for fetching the productGroup @param context @param productCode @return @	This method returns CreditPolicyTemplateDTO on product group code. @param applicationContext @param productGroupCode
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	This method is used to validate UDI. @param sessionContext @param accountId @param repaymentRecordsDTO @param repaymentStageDTO @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	This methods fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	This method first fetches the group which the given product code belongs to . then fetches the corresponding group's preferences
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	<pre>This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&lt;br&gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmountnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre> @param productId @param offerId @param currencyCode TODO @return product offerId @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupPreferences	This method is responsible for fetching the productGroup preferences @param context @param productCode @return @	This method first fetches the group which the given product code belongs to . then fetches the corresponding group's preferences
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	This methods fetches the product for the productId passed. @param context @param productId @param offerId @param currencyCode @return product LoanProduct @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	This method fetches the statement parameters for the given product code



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	<pre>This method fetches all the details of the product defined at Offer level , product level ,product group level ,statement parameters .&lt;br&gt; CurrencyCode is an optional input, If present then Currency level parameters defined for the product are also fetched. Currency level parameters defined at product preference level 1)MinimumSliptAmount , 2)LargeLoanAmountnForReporting 3)MaxLoanTermForReporting 4)SmallFeitBalance 5)SmallWaiverBalance 6)SmallLoanAmountForReporting 7)EconomicCostThreshold 8)MinArrearForReporting. currency parameters defined at offer level are 1)MinimumLoan 2)MaximumLoanAmount 3)MinimumRedrawTransactionAmount. </pre> @param productId @param offerId @param currencyCode TODO @return product offerId @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	This method fetches the statement parameters for the given product code
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	This method fetches statement parameters preferences for the given product code.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	<p>This method is responsible for updating latest Loan Statement Request which is yet to be executed.  When the Statement frequency is changed, the Adhoc Frequency Change Date of pending Statement Request will be updated with current date   (i.e. the date on which Loan Account Preferences have been modified with the change in statement frequency).    <b><u>NOTE</u></b> ::  When Customer requests for Statement facility on adhoc basis (i.e. on any Random Date),  This method derives Immediate Statement Generation Date (which must be after Current Date) and raises a "System Initiated" Statement Request. </p> @param applicationContext @param updatedPreferences
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	<p>This method is responsible for estimating <ul><li>the immediate Statement generation Date</li><li>Statement Generation Date next to Immediate Statement Generation Date</li></ul> based on the Statement Frequency provided.  While enquiring Loan Account Preferences, existing Statement Frequency is provided.  Based on this frequency, Statement Generation Dates will be estimated. </p> @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchStatementParameters	This method is responsible for fetching the Statement Parameters @param context @param productCode @return @	<p>This method is responsible for simulating the estimated "Statement Generation Dates" when the Statement Frequency has been changed at Loan Account Preferences.  It will provide the following information:  <ul> <li>Estimated Statement Generation Dates (using existing Statement Frequency) :-  [ Immediate Statement Generation Date, Next Statement Generation Date, ... ]</li> <li>Estimated Statement Generation Dates (using Changed Statement Frequency) :-  [ Immediate Statement Generation Date, Next Statement Generation Date, ... ]</li> </ul> </p> @param sessionContext @param existingStatementFrequency @param newStatementFrequency @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupDetailsOnBusinessUnit	@param appContext @param productGroupCode @return	This method fetches rate lock information. 1> fetches product group code. 2>fetches credit policy template id from product group code (calling loan product adapter method) fetches the credit template details on the basis of credit policy Template .Sets the value in LoanProductPreferenxeDto and business unit.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchCreditPolicy	@param appContext @param creditPolicyTemplateDTO @return	This method fetches rate lock information. 1> fetches product group code. 2>fetches credit policy template id from product group code (calling loan product adapter method) fetches the credit template details on the basis of credit policy Template .Sets the value in LoanProductPreferenxeDto and business unit.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchCreditPolicy	@param appContext @param creditPolicyTemplateDTO @return	This method Credit Policy Code on basis on business unit, and product group code.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchCreditPolicy	@param appContext @param creditPolicyTemplateDTO @return	This method fetchCreditPolicyTemplate fetches credit policy template details.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchCreditPolicy	@param appContext @param creditPolicyTemplateDTO @return	This method will fetch all the details of the product group. @param sessionContext @param productGroupCode
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchCreditPolicy	@param appContext @param creditPolicyTemplateDTO @return	This method returns CreditPolicyTemplateDTO on product group code. @param applicationContext @param productGroupCode
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	This method is to validate the amendConfiguration, It first validates whether the account is eligible for amend. and then  if the Amend includes offer swap then it also checks if the account is eligible for offer swap.  On submission of the configuration, The following validation happens:-   <li> <ul> Offer swap validations, if the source offer can be swapped to the target offer </ul> <ul> Amount/Term related validations </ul> <ul> Periodic Credit Instruction validations </ul> <ul> Loan Accelerated Payment Instruction validations </ul> <ul> Loan Principal Offset Validations </ul> <ul> Loan RePayment Instruction validations </ul> <ul> Loan RollOver Instruction validations </ul> <ul> Loan Schedule Related validations, if Repayment Frequency modified for any of the existing changes </ul> </li> @param loanAccountAmendConfigurationDTO @param loanAmendConfigurationParameterDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	This method does the offer swap validations @param sessionContext @param amendTypes @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	<pre>This method is used to validate the offerId expiry date. &lt;br&gt; The posting date is compared with the offerExpiry Date to check if offer is in valid period or not. &lt;br&gt; </pre> @param loanLoanOfferSwapDetailsDTO @param oldOfferId
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	This method is responsible for performing Loan Offer Swapping @param sessionContext @param offerSwapDetailsDTO
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	@param applicationContext @param loanAccount @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	This method will check if given offer id is active for current posting date.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOffer	This method will check if given offer id is active for current posting date. @param appContext @param offerId @return	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param sessionContext @param productCode @param branchCode @param currency
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOfferForTransactionType	This method will check if given transaction type is allowed for the offer.	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	validateOfferForTransactionType	This method will check if given transaction type is allowed for the offer.	@param applicationContext @param loanAccount @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchAllEligibleLoanOffers	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param applicationContext @param branchCode @param currency	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param sessionContext @param productCode @param branchCode @param currency
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchAllEligibleLoanOffers	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param applicationContext @param branchCode @param currency	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param sessionContext @param productCode @param branchCode @param currency
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchAllEligibleLoanOffers	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param applicationContext @param branchCode @param currency	This method will fetch all the offers which caters to branchCode and is defined for given currency and active for current date. @param sessionContext @param productCode @param branchCode @param currency

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupDetails	This method will fetch all the details of the product group. @param sessionContext @param productGroupCode	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupDetails	This method will fetch all the details of the product group. @param sessionContext @param productGroupCode	This method fetches rate lock information. 1> fetches product group code. 2>fetches credit policy template id from product group code (calling loan product adapter method) fetches the credit template details on the basis of credit policy Template .Sets the value in LoanProductPreferenceDto and business unit.
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchProductGroupDetails	This method will fetch all the details of the product group. @param sessionContext @param productGroupCode	<pre>This method is called to fetch fetchMainLine's productCode. The following steps are carried out in this method:- &lt;br&gt; 1)MainLine Id is fetched by calling fetchMainFacilityForAccount method of LimitsAccountApplicationService(). 2)MainLine details are fetched by calling method LoanLCMAAdapter.fetchFacilityDetails(String). 3)Then mainline's productGroupCode is fetched from the response. 4)Call is given to fetchProductGroupDetails(ApplicationContext, String) to check whether the mainline's productgroup is MOF or NOT. 5)If mainline's product group is MOF then product group code of mainline is returned. </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchAllActiveProductsForGroupCode	This method will fetch all the active products for the given Product Group Code. @param sessionContext @param productGroupCode	<pre>This method is use to fetch fetchAllowedAllowedGroupType for the given product group. The following steps are carried out in this method:- &lt;br&gt; 1)fetchAllActiveProductsForGroupCode of ILoanProductAdapter is called to fetch ProductInquiryREsponse. If response is not null ProductInformationDTO is fetched. 2)List of

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				productGroupCodes is fetched from ProductInformationDTO[] if ProductInformationDTO is not zero. </pre>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchAllActiveProductsForGroupCode	This method will fetch all the active products for the given Product Group Code. @param sessionContext @param productGroupCode	This method will fetch all the active products for the given Product Group Code. @param sessionContext @param productGroupCode
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLedgerCodeForProduct	This method is used to fetch the ledger code mapped to an accountRole and productCode from PMU module. @param sessionContext @param productCode @param accountRole @return ledgerCode @throws FatalException	Method to fetch Ledger Code based on Account Role for the event @param loanAccount @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLedgerCodeForProduct	This method is used to fetch the ledger code mapped to an accountRole and productCode from PMU module. @param sessionContext @param productCode @param accountRole @return ledgerCode @throws FatalException	This method constructs an {@link AccountingEventDTO} and raises an accounting event. NOTE:This method is specifically created for the method manageFundsOnStandingInstructionExecution as one of the account role's (LN_SI_BRIDGE) for raising the accounting event is module supplied and ledger code need to be derived for the same. @param sessionContext @param accountId @param applicationContext @param loanAccount @param amountInAccountCurrency @param loanAccountingService @return amountTagDetailsDTO @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLedgerCodeForProduct	This method is used to fetch the ledger code mapped to an accountRole and productCode from PMU module. @param sessionContext @param productCode @param accountRole	This method is used to fetch the ledger code mapped to an accountRole and productCode from PMU module. @param sessionContext @param productCode @param accountRole @return ledgerCode @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return ledgerCode @throws FatalException	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	This method is used to process late charges if any. It is called only for account opened in offer which has Basis for penalty as Penalty fee. While Raising arrear, calculate and persist tolerance amount in loan arrear. If sum of all unpaid arrear is greater than sum of their corresponding tolerance amount, then call pricing to charge penalty fee. @param context @param accountId @param actionDate
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	@param applicationContext @param accountId @param loanAccount @param servicingEventId @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	The method is responsible to fetch fees for an event in Reco mode. @param applicationContext @param accountConfigurationDTO @param loanEventCodes @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	The method is responsible to fetch fees for an event in Reco mode. @param applicationContext @param accountConfigurationDTO @param loanEventCodes @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	This methods does auto disbursement As per data provided in LoanAutoDisbursementData. @param applicationContext @param accountId @param disbursementData @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDeductionModeTypeOnEvent	This method fetched the Fee Treatment maintained for the product. @param applicationContext @param accountId @param eventId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDomainCategoryForEvent	This method fetch the accounting domain category is product based or not. @param sessionContext @param eventCode @return @throws FatalException	This method fetch the accounting domain category is product based or not. @param sessionContext @param eventCode @return boolean
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDomainCategoryForEvent	This method fetch the accounting domain category is product based or not. @param sessionContext @param eventCode @return @throws FatalException	This method calls the PMU module to fetch the offset offer details linked with a given offer. @param sessionContext @param offerId @return LoanOfferDTO @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDomainCategoryForEvent	This method fetch the accounting domain category is product based or not. @param sessionContext @param eventCode @return @throws FatalException	<p>&lt;p&gt;This method prepares Accounting Event details, which are required to log an Accounting Event for a particular Transaction.&lt;br&gt; Below details provide the information about the parameters, about which, the knowledge is required in order to prepare Accounting Event Details.&lt;br&gt; &lt;/p&gt; &lt;p&gt; &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;&lt;b&gt;Parameter Name:&lt;/b&gt; listofAmountTagDetailsBySettlement &lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter represents the list of Accounting Event - Amount tag Details. &lt;br&gt; After performing Settlement with CASA Accounts, Settlement provides Amount tag details. &lt;br&gt; "Loan Draw Down Operation" and "Loan Redraw Operation" are suitable examples to be considered for this. &lt;br&gt; &lt;br&gt; In case of Draw down, Settlement provides &lt;b&gt;Debit Amount Tag&lt;/b&gt; details. &lt;br&gt; &lt;b&gt;Credit Amount Tag&lt;/b&gt; details will be prepared by Loan Accounting Service.&lt;br&gt; &lt;br&gt; In case of Redraw, Settlement provides &lt;b&gt;Credit Amount Tag&lt;/b&gt; details. &lt;br&gt; &lt;b&gt;Debit Amount Tag&lt;/b&gt; details will be prepared by Loan Accounting Service.&lt;br&gt; &lt;br&gt; &lt;b&gt;&lt;u&gt;NOTE:&lt;/u&gt;&lt;/b&gt; If the Settlement is done with more than one CASA Accounts, the list will have more than one Amount Tag Details.&lt;br&gt; &lt;br&gt; The cases, when there is no involvement of Settlement, this</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>argument need to be passed as "NULL".&lt;br&gt; Loan Arrears Capitalization is a suitable example to be considered for this.&lt;br&gt; <b><u>NOTE:</u></b>&lt;br&gt; Both (Credit Amount Tag &amp; Debit Amount Tag) Amount Tag Details will be prepared by Loan Accounting Service.&lt;br&gt; &lt;br&gt; &lt;/li&gt;</p> <ul style="list-style-type: none"> <li>&lt;b&gt;Parameter Name:&lt;/b&gt; transactionDescription&lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter represents the description about the transaction for which Accounting Event need to be logged.&lt;br&gt; &lt;br&gt; &lt;/li&gt;</li> <li>&lt;b&gt;Parameter Name:&lt;/b&gt; amountTagToDirectionMap &lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter represents mapping between Amount Tag and the Direction of flow of amount.&lt;br&gt; &lt;br&gt; For Debit Amount Tag, the DirectionType being mapped is OUT. &lt;br&gt; For Credit Amount Tag, the DirectionType being mapped is IN. &lt;br&gt; This parameter plays an important role while populating the Amount tag details.&lt;br&gt; &lt;br&gt; &lt;/li&gt;</li> <li>&lt;b&gt;Parameter Name:&lt;/b&gt; isCashTransaction&lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter identifies the instrument reference sent by the module required for clearing/payment reconciliation.&lt;br&gt; Here, it needs to be passed as NULL.&lt;br&gt; &lt;br&gt; &lt;/li&gt;</li> <li>&lt;b&gt;Parameter Name:&lt;/b&gt; nonProductDomainCategory &lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter indicates the event (for which Accounting Event need to be logged) &lt;br&gt; has occurred due to Non Product Domains (like Payment, Cash Transaction, ATM, Bill Payment etc). Above mentioned examples are non-products, or Product has not been defined for above categories.&lt;br&gt; &lt;br&gt; &lt;/li&gt;</li> <li>&lt;b&gt;Parameter Name:&lt;/b&gt; isTransactionFromLoanAccountToExternalAccount&lt;br&gt; &lt;b&gt;Description:&lt;/b&gt; This parameter identifies</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>whether amount transfer is from Loan Account to CASA Accounts or vice versa.&lt;br&gt; If the transaction flow is Loan Account -&gt; CASA Account (e.g. Redraw), this parameter need to be passed as TRUE. &lt;br&gt; If the transaction flow is CASA Account -&gt; Loan Account (e.g. Drawdown), this parameter need to be passed as FALSE. &lt;br&gt; &lt;/li&gt; &lt;/ol&gt; &lt;/p&gt;           @param applicationContext @param listOfAmountTagDetailsBySettlement @param loanAccountId @param eventCode @param transactionDescription @param amountTagToDirectionMap @param isCashTransaction @param instrumentId @param transAmountInAccountCurrency @param nonProductDomainCategory @param isTransactionFromLoanAccountToExternalAccount @return AccountingEventDTO</p>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchOfferOffsetLinkage	This method calls the PMU module to fetch the offset offer details linked with a given offer. @param sessionContext @param offerId @return LoanOfferDTO @throws FatalException	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchOfferOffsetLinkage	This method calls the PMU module to fetch the offset offer details linked with a given offer. @param sessionContext @param offerId @return LoanOfferDTO @throws FatalException	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchOfferOffsetLinkage	This method calls the PMU module to fetch the offset offer details linked with a given offer. @param sessionContext @param offerId @return LoanOfferDTO @throws FatalException	This method calls the PMU module to fetch the offset offer details linked with a given offer. @param sessionContext @param offerId @return LoanOfferDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU Setup. @param productId @return	<p>This method is responsible for logging the Delinquency Details of a Loan Account. It will ADD / UPDATE the delinquency counter bucket when arrears are raised.</p> <p><b>NOTE:</b> This method must to be invoked only when the Arrear is being raised for the first time.</p> <p><b>APPROACH</b></p> <ul style="list-style-type: none"> <li>Fetch Delinquency Day Ranges from Product Manufacturing Unit</li> <li>Fetch Delinquency Counters</li> <li>Delete existing Delinquency Counters from Session</li> <li>Fetch Unpaid Arrears</li> <li>Prepare Delinquency Counters details using following elements:- <ul style="list-style-type: none"> <li>Existing Delinquency Counters details</li> <li>Delinquency Day Ranges</li> <li>Unpaid Arrears</li> <li>Arrears being raised at present</li> </ul> </li> <li>Save updated Delinquency Counters details</li> <li>Prepare and save Delinquency Counter History for every single Delinquency Counter record.</li> <li>Update Delinquency Counters details in Aggregates for the Loan Account.</li> <li>Calculate "Action Due Date" for next Bucket Movement.</li> </ul> <p>Action Due Date will be calculated as:</p> <p><b>EXCLUSIVE Day Ranges :-</b></p> <p>Action Date := Arrears Due Date + (Delinquency Day Range - 1)</p> <p>e.g. Arrears are raised on 31st January 2012 (Leap Year) Range Days: 30 Then, Action Date for Bucket Movement: 29th February 2012.</p> <p><b>INCLUSIVE Day Ranges :-</b></p> <p>Action Date := Arrears Due Date + Delinquency Day Range</p> <p>e.g. Arrears are raised on 31st January 2012 (Leap Year) Range Days: 30 Then, Action Date for Bucket Movement: 1st March 2012.</p> <ul style="list-style-type: none"> <li>Log the action for Delinquency Counter - Bucket Movement.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param LoanArrear
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU SetUp. @param productId @return	<p>&lt;p&gt;This method is responsible for logging the Delinquency Details of a Loan Account for backdated transaction.&lt;br&gt; &lt;br&gt;</p> <p>&lt;b&gt;&lt;u&gt;NOTE:&lt;/u&gt;&lt;/b&gt;&lt;br&gt;This method must to be invoked only when the backdated transaction is performed.&lt;br&gt; &lt;br&gt; &lt;u&gt;APPROACH &lt;/u&gt; &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch Delinquency Day Ranges from Product Manufacturing Unit&lt;/li&gt;</li> <li>&lt;li&gt;Fetch Unpaid Arrears&lt;/li&gt;</li> <li>&lt;li&gt;Prepare Delinquency Counters details using following elements:- &lt;ul&gt;</li> <li>&lt;li&gt;Delinquency Day Ranges&lt;/li&gt;</li> <li>&lt;li&gt;Unpaid Arrears&lt;/li&gt;</li> </ul> </li> <li>&lt;li&gt;Save updated Delinquency Counters details&lt;/li&gt;</li> <li>&lt;li&gt;Prepare and save Delinquency Counter History for every single Delinquency Counter record.&lt;/li&gt;</li> <li>&lt;li&gt;Calculate "Action Due Date" for next Bucket Movement.&lt;br&gt;</li> <p>&lt;br&gt; Action Due Date will be calculated as:&lt;br&gt; &lt;br&gt; &lt;u&gt;EXCLUSIVE Day Ranges :-&lt;/u&gt;&lt;br&gt; &lt;b&gt;Action Date := Arrears Due Date + (Delinquency Day Range - 1)&lt;/b&gt; &lt;br&gt; e.g. Arrears are raised on 31st January 2012 (Leap Year) &lt;br&gt; Range Days: 30 &lt;br&gt; Then, Action Date for Bucket Movement: 29th February 2012.</p> <p>&lt;br&gt; &lt;u&gt;INCLUSIVE Day Ranges :-&lt;/u&gt;&lt;br&gt; &lt;b&gt;Action Date := Arrears Due Date + Delinquency Day Range&lt;/b&gt; &lt;br&gt; e.g. Arrears are raised on 31st January 2012 (Leap Year) &lt;br&gt; Range Days: 30 &lt;br&gt; Then, Action Date for Bucket Movement: 1st March 2012. &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Log the action for Delinquency Counter - Bucket Movement.&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; &lt;/p&gt;</p> <p>@param accountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU SetUp. @param productId @return	<p>This method is responsible for updating Delinquency Counters</p> <ul style="list-style-type: none"> <li>when arrears are <b>paid</b>.</li> <li>when arrears are <b>capitalized</b>.</li> <li>when raised arrears are <b>roll-backed / removed (during Reversal)</b>.</li> <li>when arrears are <b>written-off</b> completely.</li> </ul> <p><b>NOTE:</b> Place a call for this method in following locations:</p> <ul style="list-style-type: none"> <li>LoanArrearService -&gt; pay()</li> <li>LoanArrearService -&gt; capitalizeUnpaidArrears() [before this.capitalize()]</li> <li>LoanArrearService -&gt; reverseArrear()</li> <li>ArrearService -&gt; updateArrearToWriteOff()</li> </ul> <p>A HashMap will be maintained with (Arrears Subsequence Number / Arrears Due Date) as Key / Value pair.</p> <p><b>Approach:</b></p> <ul style="list-style-type: none"> <li>Fetch all Delinquency details for provided Loan Account</li> </ul> <p><b>CALL:</b> LoanDelinquencyCounterRepository -&gt; fetchAllDelinquencyDetails()</p> <ul style="list-style-type: none"> <li>For current Arrears (which is being paid)</li> <li>Fetch paid arrears for the Loan Account which were raised for the provided due Date.</li> </ul> <p><b>CALL:</b> fetchPaidArrearsRaisedOnDay(accountId, LoanArrear -&gt; dueDate)</p> <ul style="list-style-type: none"> <li>IF list is empty THEN</li> <li>BEGIN</li> <li>IF HashMap don't have record for Arrears THEN</li> <li>Identify the Bucket (which holds the arrears reference number of the current arrears) from the list</li> <li>Remove Arrears Sequence Number from LoanDelinquencyDetails</li> <li>Decrement Delinquency Counter by 1</li> <li>Update the list of delinquency details.</li> <li>Add the entry for the current arrears (arrearRefNo, dueDate) in HashMap.</li> <li>END</li> <li>IF list is empty THEN</li> <li>BEGIN</li> <li>IF HashMap have record for Arrears THEN</li> <li>BEGIN</li> <li>Identify the Bucket (which holds</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the arrears reference number of the current arrears) from the list &lt;br&gt; Remove Arrears Sequence Number from LoanDelinquencyDetails &lt;br&gt; Update the list of delinquency details. &lt;br&gt; END &lt;br&gt; END &lt;br&gt; &lt;br&gt; IF list is not empty THEN &lt;br&gt; Identify the Bucket (which holds the arrears reference number of the current arrears) from the list &lt;br&gt; Remove Arrears Sequence Number from LoanDelinquencyDetails &lt;br&gt; Update the list of delinquency details.&lt;/li&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;NOTE: If delinquency Counter = 0 THEN</li> <li>&lt;br&gt;Delete the respective bucket. &lt;br&gt; &lt;/li&gt;</li> <li>&lt;li&gt;Save the updated delinquency details &lt;br&gt; CALL: logDelinquencyDetailsForPaidArrears() for every Delinquency detail&lt;/li&gt;</li> <li>&lt;li&gt;Update Delinquency Details in Loan Account Aggregate.&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; &lt;/p&gt;</p> <p>@param arrearBeingPaid</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU SetUp. @param productId @return	<p>This method is responsible for restoring the Delinquency Counters for a Loan Account. It will ADD / UPDATE the delinquency counter bucket when arrears are restored.</p> <p><b>NOTE:</b> This method must to be invoked only when the Arrear is being restored.</p> <p><b>TECHNIQUE for Restoring Delinquency Counters:</b></p> <ul style="list-style-type: none"> <li><b>CASE I :: Reversal of Paid Arrears</b> In this scenario, Delinquency Counters will be logged for the restored arrears. Day Range Bucket will be determined based on no. of Days passed between Arrears Due Date and Current Date. A bucket movement action will be logged for the same again. Refer :: <a href="#">{@link LoanDelinquencyCounterManagementService#restoreDelinquencyCountersOnTransactionReversal(LoanArrear)}</a></li> <li><b>CASE II :: Reversal of Raised Arrears</b> In this scenario, Delinquency Counters will be removed for the reversal of raised arrears. This is similar to "Removing Delinquency Counters" on payment of arrears. Refer :: <a href="#">{@link LoanDelinquencyCounterManagementService#updateDelinquencyCounter(LoanArrear)}</a></li> </ul> <p><b>APPROACH</b></p> <ul style="list-style-type: none"> <li>Fetch Delinquency Day Ranges from Product Manufacturing Unit</li> <li>Fetch Delinquency Counters</li> <li>Delete existing Delinquency Counters from Session</li> <li>Fetch Unpaid Arrears</li> <li>Prepare Delinquency Counters details using following elements:- <ul style="list-style-type: none"> <li>Existing Delinquency Counters details</li> <li>Delinquency Day Ranges</li> <li>Unpaid Arrears</li> <li>Arrears being raised at present</li> </ul> </li> <li>Save updated</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Delinquency Counters details&lt;/li&gt; &lt;li&gt;Prepare and save Delinquency Counter History for every single Delinquency Counter record.&lt;/li&gt; &lt;li&gt;Update Delinquency Counters details in Aggregates for the Loan Account.&lt;/li&gt; &lt;li&gt;Calculate "Action Due Date" for next Bucket Movement.&lt;br&gt; Action Due Date will be calculated as:&lt;br&gt; &lt;u&gt;EXCLUSIVE Day Ranges :-&lt;/u&gt;&lt;br&gt; &lt;b&gt;Action Date := Arrears Due Date + (Delinquency Day Range - 1)&lt;/b&gt; &lt;br&gt; e.g. Arrears are raised on 31st January 2012 (Leap Year) &lt;br&gt; Range Days: 30 &lt;br&gt; Then, Action Date for Bucket Movement: 29th February 2012. &lt;br&gt; &lt;u&gt;INCLUSIVE Day Ranges :-&lt;/u&gt;&lt;br&gt; &lt;b&gt;Action Date := Arrears Due Date + Delinquency Day Range&lt;/b&gt; &lt;br&gt; e.g. Arrears are raised on 31st January 2012 (Leap Year) &lt;br&gt; Range Days: 30 &lt;br&gt; Then, Action Date for Bucket Movement: 1st March 2012. &lt;br&gt; &lt;/li&gt;</p> <p>&lt;li&gt;Log the action for Delinquency Counter - Bucket Movement.&lt;/li&gt; &lt;/ol&gt; &lt;/p&gt; @param arrearBeingRestored</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU Setup. @param productId @return	<p>This method is responsible for identifying the Nature of Delinquency Day Ranges defined as a part of Product Setup.&lt;br&gt; Response of this method will decide the mechanism of calculating Next Action Date for Bucket Movement.&lt;br&gt; &lt;br&gt; &lt;u&gt;Inclusive Day Ranges :-&lt;/u&gt;&lt;br&gt; 0 - 7   8 - 29   30 - 59   60 - 89   90 - 119   120 - 99999&lt;br&gt; Action Due Date will be calculated as: &lt;br&gt; Action Date := Arrears Due Date + Delinquency Day Range &lt;br&gt; &lt;br&gt; &lt;u&gt;Exclusive Day Ranges :-&lt;/u&gt;&lt;br&gt; 0 - 10   10 - 30   30 - 60   60 - 90   90 - 180   180 - 99999&lt;br&gt; Action Due Date will be calculated as: &lt;br&gt; Action Date := Arrears Due Date + (Delinquency Day Range - 1) &lt;br&gt; &lt;br&gt; &lt;b&gt;NOTE:&lt;/b&gt;&lt;br&gt; If current Delinquency Day Range is not available / not provided then minimum Day Range will be derived.&lt;br&gt; Thus, if "currentDayRange" = 0, the method will derive minimum Day Range. &lt;/p&gt; @param productId @param currentDayRange @param delinquencyDayRanges @return</p>
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU Setup. @param productId @return	<p>This method is defined to fetch Delinquency Counter History items.&lt;br&gt; If the boundary dates are provided (i.e. From Date and To Date), then it will filter Delinquency Counter History items based on the dates provided.&lt;br&gt; &lt;ul&gt; &lt;b&gt;&lt;u&gt;Filter Criteria&lt;/u&gt;&lt;/b&gt; &lt;li&gt;whose Posting Date is after or equal to provided "FROM DATE"&lt;/li&gt; &lt;li&gt;whose Posting Date is before or equal to provided "TO DATE"&lt;/li&gt; &lt;/ul&gt; If "FROM DATE" and "TO DATE" being provided have inappropriate values like &lt;ul&gt; &lt;li&gt;01/01/1800 (Considered as Blank Date / Default Date)&lt;/li&gt; &lt;li&gt;01/01/1950 (Considered as Null Date)&lt;/li&gt; &lt;/ul&gt; Then entire delinquency Counter history for the Loan Account will be derived. &lt;/p&gt; @param accountId @param fromDate @param toDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU SetUp. @param productId @return	<p>This method is exclusively defined to show overdue payments for the provided Loan Account ID for Loan Serve.  <ul> <b>Information provided about Overdue Payments</b> <li>Delinquency Day Range: Day Range Start - Day Range End</li> <li>No. of Days for Overdue (i.e. Delinquency Counters)</li> </ul> </p> @param sessionContext @param accountId @return @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU SetUp. @param productId @return	<pre>This method is responsible for initiating the process of "Loan Delinquency Details Update". NOTE Capture a Due Date for Reference only once. This Due Date need to be compared with every single arrears which qualifies above criteria. This will help in identifying Arrears raised on same Date. For them, Delinquency Counter won't be decremented / incremented. A HashMap will be maintained with (Arrears Subsequence Number / Arrears Due Date) as Key / Value pair. Approach:- 1) Get the list of Unpaid Arrears for the Loan Account. 2) Get the list of Delinquency Day Ranges from Product 3) Get list of Delinquency Counter Details for the respective Loan Account. 4) Get the "Next Delinquency Day Range" from Product. 5) Identify whether Delinquency Day ranges are INCLUSIVE / EXCLUSIVE. 6) Filter Arrears which are due for Bucket Movement by applying following criteria. CASE I :: EXCLUSIVE Day Range ----- IF ( LoanArrear -> dueDate + LonaDelinquencyDetails -> delinquencyDayRange) EQUALS (actionDate+1) THEN Arrear is due for Bucket Movement. CASE II :: INCLUSIVE Day Range ----- IF ( LoanArrear -> dueDate + LonaDelinquencyDetails -> delinquencyDayRange) EQUALS (actionDate) THEN

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Arrear is due for Bucket Movement. 7) Maintain two types of Mappings :- A) removedArrearsMap This map will have distinct entries of Removed Arrears for a Due Date It will help in decrementing the Counter only once when multiple arrears with same due date are removed from a bucket. B) addedArrearsMap This map will have distinct entries of Added Arrears for a Due Date It will help in incrementing the Counter only once when multiple arrears with same due date are added to next bucket. 8) Bucket Movement Process :- ----- For every Unpaid Arrear which is due for Bucket Movement IF listOfFilteredUnpaidArrears is not empty THEN REMOVE ARREAR FROM PRESENT BUCKET IF (removedArrearsMap) HashMap is empty THEN - Add the entry for the current arrear (arrearRefNo, dueDate) in HashMap. -Identify the Bucket (which holds the arrears reference number of the current arrears) from the list (i.e. CHECK IF LoanDelinquencyDetails -&gt; arrearReferenceNumbers CONTAINS( LoanArrear -&gt; ArreayKey -&gt; subSequenceNumber)) -Remove Arrears Sequence Number from LoanDelinquencyDetails -Decrement Delinquency Counter by 1 -IF Delinquency Counter IS 0 THENDelete the Bucket Details ELSE Update delinquency details in the list. ELSE IF (removedArrearsMap) HashMap is not empty, and has an entry of an Arrear for a Due Date THEN - Identify the Bucket (which holds the arrears reference number of the current arrears) from the list (i.e. CHECK IF LoanDelinquencyDetails -&gt; arrearReferenceNumbers CONTAINS( LoanArrear -&gt; ArreayKey -&gt; subSequenceNumber)) -Remove Arrears Sequence Number from LoanDelinquencyDetails ELSE IF</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>(removedArrearsMap) HashMap is not empty, and doesn't has an entry of an Arrear for a Due Date THEN  -Add the entry for the current arrear (arrearRefNo, dueDate) in HashMap. -Identify the Bucket (which holds the arrears reference number of the current arrears) from the list -Remove Arrears Sequence Number from LoanDelinquencyDetails -Update the list of delinquency details. -IF Delinquency Counter IS 0 THENDelete the Bucket Details ELSE Update delinquency details in the list. ADD ARREAR TO NEXT BUCKET IF Bucket for Next Day Range exists THEN IF (addedArrearsMap) HashMap is empty THEN -Add the entry for the current arrear (arrearRefNo, dueDate) in HashMap. -Append Arrears Sequence Number to LoanDelinquencyDetails -Increment Delinquency Counter by 1 -Update delinquency details in the list. ELSE IF (addedArrearsMap) HashMap is not empty and has an entry of an Arrear for a Due Date THEN -Append Arrears Sequence Number to LoanDelinquencyDetails -Update delinquency details in the list. ELSE IF (addedArrearsMap) HashMap is not empty, and doesn't has an entry of an Arrear for a Due Date THEN -Add the entry for the current arrear (arrearRefNo, dueDate) in HashMap. -Append Arrears Sequence Number to LoanDelinquencyDetails -Increment Delinquency Counter by 1 -Update delinquency details in the list. ELSE (i.e. No Bucket exists for Next Day Range) -Prepare Fresh Delinquency details for Next Bucket. -Set Loan Account ID -Set Arrears Reference Number -Set Delinquency Day Range with the value of Next Day Range -Set Delinquency Counter as 1 9) Call  LoanDelinquencyCounterManagementService -&gt; updateDelinquencyCounterOnBucketMovement(List</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				of Delinquency Details, Next Delinquency Day Range) </pre> @param applicationContext @param accountId @param actionDate
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU Setup. @param productId @return	<p>This method is responsible for fetching Delinquency Day ranges defined at PMU Setup.  Delinquency Day ranges will be sorted in ascending order of Day Range End. @param productId @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDelinquencyDayRanges	This method is responsible for fetching Delinquency Day ranges defined at PMU Setup. @param productId @return	
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchDDABankPolicy		
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		@param accountId @param backValueDate @param eventCode @param maintenanceType @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		@param accountId @param bankCode @param effectiveDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		1.Bank policy definition should allow backdated transaction (This check is only done for loans) 2.Account opening date i.e. the backdated settlement payout date should not be before the backdated term specified in the same   3.Account opening date i.e. the backdated settlement payout date should not be before the offer acceptance date @param accountOpeningDate @param offerAcceptanceDate @throws FatalException
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		This method returns the {@link StatementLogType} @param applicationContext @return StatementLogType
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchLoanBankPolicy		This method checks the rollover Date. The rollover date should be greater than the greatest of Account opening date, Last disbursement date, Last Rate Change Date, Last Partial Payoff Date, Last deferment date, Last Repayment Review /Rescheduling date @param accountId



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchRepriceEventsForBankCode		This method performs the operation of checking whether the account needs to be re-Priced for the given eventId. @param applicationContext @param accountId @param eventId @param onDate @return
ILoanProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProductAdapter	fetchRepriceEventsForBankCode		
IPCCashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCashAdapter	updateCash	@param componentPreferenceDTO @param referenceNumber @param transactionAmount @throws FatalException	
IPCCashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCashAdapter	updateCash	@param componentPreferenceDTO @param referenceNumber @param transactionAmount @throws FatalException	This method will update the cash position in the broput module
IPCCashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCashAdapter	updateCash	@param componentPreferenceDTO @param referenceNumber @param transactionAmount @throws FatalException	This service will do all the processing if the settlement mode is CASH in the bank draft issue. <li>It will do the amount conversion based on the currency used.(cross currency transaction). <li>It will validate the converted amount to the instrument amount(based amount) and will prompt an error in case of mismatch in amounts. @param transactionContext @param iBankDraft @param featureDTO @param settlementModeType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPaymentsAdapter	isPendingTransactionAvailable	This method returns TransactionStatus true if there are no pending Payments on the Account. If Pending Payments are present on the account, the method throws a Business exception @param accountId Term Deposit Account Id @param moduleType Module Type @return boolean @throws FatalException	This method will close the account i.e. account will be marked as Closed @param sessionContext @param termDepositAccount @return TransactionStatus @throws FatalException
ITDPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPaymentsAdapter	fetchInstitutionDetailsForCounterPartyAccount	This will fetch the institution id and institution type for the given external account number. @param accountNumber @param partyId TODO @return @throws FatalException	This is the light API which is used to inquire the redemption details and will perform the actual redemption. This method will be called by caller who want to settle the amount to external account.  Term deposit will be redeemed and it will call settlement. Also it will raise accounting events  This API accepts account id, deposit number and redemption amount along with the external account id  @param accountId Account id of term deposit @param depositId Term deposit number @param redemptionAmount Money object containing redemption amount @param externalAcctId external account number @param remitterName Remitter name @param redemptionAmountType Type of redemption whether full or part @return {@link RedemptionResponseDetailsDTO} Object containing result of inquiry and redemption operation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPaymentsAdapter	fetchInstitutionDetailsForCounterPartyAccount	This will fetch the institution id and institution type for the given external account number. @param accountNumber @param partyId TODO @return @throws FatalException	This is a service which allows payin through cheque drawn on an external account. The TD related information is derived from the TD product definition based on the account ID given. The other input that is mandatory needs to be supplied in the input. @param tdAccountId The account Id of the Term Deposit account (Mandatory) @param draweeAccountId The external account id (Mandatory) @param chequeNo The cheque number (Mandatory) @param chequeDate The cheque date (Mandatory) @param payinAmount The actual amount to be paid in to the deposit @param term The term for which the deposit is to be created. This is optional, if null the minimum term from the product definition will be taken. (Optional) @return @throws FatalException
ITDPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPaymentsAdapter	fetchInstitutionDetailsForCounterPartyAccount	This will fetch the institution id and institution type for the given external account number. @param accountNumber @param partyId TODO @return @throws FatalException	This method provides an api for TD payin via an external account. The TD specific information will be derived from the TD product definition. @param accountId The account Id of the Term Deposit account (Mandatory) @param providerAccountId The external account id (Mandatory) @param payinAmount The actual amount to be paid in to the deposit @param term The term for which the deposit is to be created. This is optional, if null the minimum term from the product definition will be taken. (Optional) @param creditorName The TD account nick name @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPayment sAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDPaymentsAdapt er	fetchInstitut ionDetailsF orCounterP artyAccoun tByCounter PartyName	This will fetch the institution id and institution type for the given counter party name. @param counterPartyName counter party name for which institution details need to be fetched @param partyId @return {@link FinancialInstitutionIdentificationDTO} @throws FatalException	This is the light API which is used to inquire the redemption details and will perform the actual redemption. This method will be called by caller who want to settle the amount to external account.  Term deposit will be redeemed and it will call settlement. Also it will raise accounting events  This API accepts account id, deposit number and redemption amount along with the external account id  @param accountId Account id of term deposit @param depositId Term deposit number @param redemptionAmount Money object containing redemption amount @param externalAcctId external account number @param remitterName Remitter name @param redemptionAmountType Type of redemption whether full or part @return {@link RedemptionResponseDetailsDTO} Object containing result of inquiry and redemption operation
ITDPayment sAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDPaymentsAdapt er	fetchInstitut ionDetailsF orCounterP artyAccoun tByCounter PartyName	This will fetch the institution id and institution type for the given counter party name. @param counterPartyName counter party name for which institution details need to be fetched @param partyId @return {@link FinancialInstitutionIdentificationDTO} @throws FatalException	This is a service which allows payin through cheque drawn on an external account. The TD related information is derived from the TD product definition based on the account ID given. The other input that is mandatory needs to be supplied in the input. @param tdAccountId The account Id of the Term Deposit account (Mandatory) @param draweeAccountId The external account id (Mandatory) @param chequeNo The cheque number (Mandatory) @param chequeDate The cheque date (Mandatory) @param payinAmount The actual amount to be paid in to the deposit @param term The term for which the deposit is to be created. This is optional, if null the minimum term from the product definition will be taken. (Optional) @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPaymentsAdapter	fetchInstitutionDetailsForCounterPartyAccountByCounterPartyName	This will fetch the institution id and institution type for the given counter party name. @param counterPartyName counter party name for which institution details need to be fetched @param partyId @return {@link FinancialInstitutionIdentificationDTO} @throws FatalException	This method provides an api for TD payin via an external account. The TD specific information will be derived from the TD product definition. @param accountId The account Id of the Term Deposit account (Mandatory) @param providerAccountId The external account id (Mandatory) @param payinAmount The actual amount to be paid in to the deposit @param term The term for which the deposit is to be created. This is optional, if null the minimum term from the product definition will be taken. (Optional) @param creditorName The TD account nick name @return @throws FatalException
ITDTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	The method de-register the deposit from Taxation module. @param deposit {@link Deposit} The term deposit for unregister

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#registerWithTax(com.ofss.fc.app.adapter.td.dto.TDTaxRegistrationDTO, com.ofss.fc.enumeration.taxation.ArrangementStatus)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc. @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#reverseRedemption(java.lang.String, java.lang.String)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration.	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module.   @param deposit {@link DepositDTO} object containing information about deposit for which interest plan needs to be registered.
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	evaluateAndUpdateTaxWaiver	This method will be used to update the tax waivers if any for the deposit This method will be called after creation of each new deposit.  Currently this method has no implementation.  Customization team can implement this method in order to apply tax waivers for a newly created deposit. @param sessionContext {@link SessionContext} Which contains basic information about the user of the system. @param accountId Identifier of account to which deposit belongs, for which waiver needs to be updated @param depositId Identifier of deposit, for which waiver needs to be updated	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module.   @param deposit {@link DepositDTO} object containing information about deposit for which interest plan needs to be registered.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	reverseRedemption	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc.	This method reverses the redemption transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	reverseRedemption	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc.	This method reverses the renewal transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	reverseRedemption	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc.	The method calls the tax module to reverse the sequence of operations done in tax due to redemption.  @param internalReferenceNumber Holds the transactionReferenceNumber of the current transaction. @param originalTransactionReferenceNumber Holds the transactionReferenceNumber of the transaction, that is meant to be reversed.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	reverseRedemption	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc.	The method calls the interest module to reverse the sequence of operations done in interest due to redemption. The method also take care of the back dated interest reversal also if the redemption was premature full. @param deposit {@link Deposit} holds the attributes of deposit entity @param lastTransactiondate Holds the date on which the last transaction was performed @param originalTransactionReferenceNumber Holds the transactionReferenceNumber of the transaction, that is meant to be reversed. @param redemptionType {@link RedemptionType} tells the type of redemption
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	reverseRedemption	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc.	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc. @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#reverseRedemption(java.lang.String, java.lang.String)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	fetchAlternateAccountForTax	This method will be used to fetch the recovery account number for an arrangement number. This method will internally call {@link this#fetchSettlementReferenceNumber()} to fetch the reference number by passing arrangement number . This reference number will be passed to settlement module to fetch the recovery account details. @param arrangementNumber A string of accountId and depositId separated by '#' @return String recovery account number	This method is used for the inquiry of the deposit attributes. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId {@link DepositKeyDTO} This parameter holds the account id. @param depositId {@link DepositKeyDTO} This parameter holds the deposit id @return DepositAttributeInquiryResponse {@link DepositAttributeInquiryResponse} @throws FatalException {@link FatalException} holds the exception coming from the transaction, if any

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	fetchAlternateAccountForTax	This method will be used to fetch the recovery account number for an arrangement number. This method will internally call <a href="#">this#fetchSettlementReferenceNumber()</a> to fetch the reference number by passing arrangement number . This reference number will be passed to settlement module to fetch the recovery account details. @param arrangementNumber A string of accountId and depositId separated by '#' @return String recovery account number	This method will be used to fetch the recovery account number for an arrangement number. This method will internally call <a href="#">this#fetchSettlementReferenceNumber()</a> to fetch the reference number by passing arrangement number . This reference number will be passed to settlement module to fetch the recovery account details. @param arrangementNumber A string of accountId and depositId separated by '#' @return String recovery account number com.ofss.fc.app.adapter.td.ITDTaxationAdapter#fetchAlternateAccountForTax(java.lang.String, java.lang.Integer)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	hasTaxDeducted	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean	This method check whether contract modification is allowed or not  1. Call <a href="#">hasContractModifiedToday()</a> <a href="#">DepositContractModificationBusinessPolicy#hasContractModifiedToday()</a>   2. Call <a href="#">hasWindowExpired()</a> <a href="#">DepositContractModificationBusinessPolicy#hasWindowExpired()</a>   3. Call <a href="#">hasBlock()</a> <a href="#">DepositContractModificationBusinessPolicy#hasBlock()</a>   4. Call <a href="#">hasTaxDeducted()</a> <a href="#">DepositContractModificationBusinessPolicy#hasTaxDeducted()</a>   5. Call <a href="#">validateTermWithLienExpiry()</a> <a href="#">DepositContractModificationBusinessPolicy#validateTermWithLienExpiry()</a> @param deposit @return boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	hasTaxDeducted	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean	Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param deposit @return boolean
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	hasTaxDeducted	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean	Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted. 
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	hasTaxDeducted	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#hasTaxDeducted(java.lang.String, java.lang.Integer)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	saveTaxState	The method saves the tax state that would be helpful in case the redemption reversal is performed. @param accountId Term deposit account id @param depositId Term deposit id @param transactionReferenceNumber Transaction reference number required to log tax state @param partyId TODO	The method saves the transaction state of a transaction.  This state would be maintained if reversal id performed on the transaction. @param accountId This attribute denotes a label representing one or more numbers of codes by which an account may be uniquely identified. These identifiers may be industry recognized identifiers or internally generated identifiers. The identifier is generated when the account relationship is first established and it never undergoes a change after that. @param depositId This deposit number represents a unique deposit in the account.  <p> The deposit number is incremented every time a new deposit is created in an account. It may or may not be created due to interest payout or renewal.  @param transactionReferenceNumber Holds the transactionReferenceNumber of the current transaction. @param sequenceNumber Holds the transactionSequenceNumber of the current transaction. @param tdTransactionType Holds the value to identify the type of transaction. @param isTransactionReversalPossible Will tell if the reversal is possible for the current transaction. @param taskCode Hold task code of the current transaction. @return {@link TransactionState}
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	saveTaxState	The method saves the tax state that would be helpful in case the redemption reversal is performed. @param accountId Term deposit account id @param depositId Term deposit id @param transactionReferenceNumber Transaction reference number required to log tax state @param partyId TODO	The method saves the tax state that would be helpful in case the redemption reversal is performed. @param accountId Term deposit account id @param depositId Term deposit id @param transactionReferenceNumber transaction reference number required to log tax state @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#saveTaxState(java.lang.String, java.lang.Integer, java.lang.String, String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	saveTaxState	The method saves the tax state that would be helpful in case the redemption reversal is performed. @param accountId Term deposit account id @param depositId Term deposit id @param transactionReferenceNumber Transaction reference number required to log tax state @param partyId TODO	This method fetches the Tax Details for Particular Deposit within a given range @param accountId Term deposit account id @param depositId Term deposit id @param fromDate Records need to be fetched having date after this @param toDate Records need to be fetched having date before this @return {@link TaxationAccountInquiryDetailsDTOResponse} Object containing taxation details of account and transaction @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#inquireTaxDetails(java.lang.String, int, com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	inquireTaxDetails	This method fetches the Tax Details for Particular Deposit within a given range @param accountId Term deposit account id @param depositId Term deposit id @param fromDate Records need to be fetched having date after this @param toDate Records need to be fetched having date before this @return {@link TaxationAccountInquiryDetailsDTOResponse} Object containing taxation details of account and transaction	This method generates the {@link DepositBalanceDetailInquiryDTO} from the {@link Deposit} which will be set as attribute in {@link DepositAttributeDTO} @param deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	inquireTaxDetails	This method fetches the Tax Details for Particular Deposit within a given range @param accountId Term deposit account id @param depositId Term deposit id @param fromDate Records need to be fetched having date after this @param toDate Records need to be fetched having date before this @return {@link TaxationAccountInquiryDetailsDTO Response} Object containing taxation details of account and transaction	This method returns true if tax has been deducted.  Tax may be deducted at accrual, compounding or capitalization.   If any such tax event has occurred on the subject deposit, contract modification of that deposit will not be permitted.  @param accountId Term deposit account id @param depositId Term deposit id @return boolean @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#hasTaxDeducted(java.lang.String, java.lang.Integer)
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	inquireTaxDetails	This method fetches the Tax Details for Particular Deposit within a given range @param accountId Term deposit account id @param depositId Term deposit id @param fromDate Records need to be fetched having date after this @param toDate Records need to be fetched having date before this @return {@link TaxationAccountInquiryDetailsDTO Response} Object containing taxation details of account and transaction	This method fetches the Tax Details for Particular Deposit within a given range @param accountId Term deposit account id @param depositId Term deposit id @param fromDate Records need to be fetched having date after this @param toDate Records need to be fetched having date before this @return {@link TaxationAccountInquiryDetailsDTO Response} Object containing taxation details of account and transaction @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#inquireTaxDetails(java.lang.String, int, com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDTaxationAdapter	updateAccountBranch	This method will be used to update the account branch code while transferring the account from one branch to another.   This will internally call the <a href="#">TaxManagementService#updateAccountBranch(DepositDetailsKey, String, String)</a> to update the branch. @param partyID @param accountID @param existingBranch @param newBranch @return @throws FatalException	This method updates the branch for the account and all of its regular deposits. This method when called via branch process. Then the deposit array and the termDepositAccount Entities are null. @param accountId @param newBranchId @param branchTransferDate @param Deposit [] <a href="#">Deposit</a> @param termDepositAccount <a href="#">TermDepositAccount</a> The method first validates the branch transferDate, The branch transfer date should be Greater than or equal to posting date and then it checks if the account is eligible for branch transfer  if the account is eligible for the transfer then the execution of the branch transfer happens by following the below steps <ul> <li>The list of deposits for the account is fetched by calling <a href="#">DepositService</a> : fetchAllDeposits() method</li> <li>Iterate through the deposit list and pick those deposits that has deposit status as regular</li> <ol> <li>update the branchId of the deposit by calling updateDepositBranch() method</li> <li>raise an Accounting Event by calling <a href="#">TDAccountingAdapter</a> : raiseAccountingEvent</li> <li>intimate the branch id change to Interest by calling <a href="#">TDInterestAdapter</a> : updateDepositBranchInInterestPlanItem</li> <li>intimate the branch change to tax by calling <a href="#">TDTaxationAdapter</a> : updateDepositBrachInTax() </li> </ol> </li> <li>finally after looping through the deposit list and updating the deposits  raise an alert by logging a nonfinancial Event by calling <a href="#">TDEventLogService</a> :logNonFinancialEvent()</li> <li>update the branch for the account by calling the updateAccountBranch</li> </ul> @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	updateAccountBranch	This method will be used to update the account branch code while transferring the account from one branch to another.   This will internally call the {@link TaxManagementService#updateAccountBranch(DepositDetailsKey, String, String)} to update the branch . @param partyID @param accountID @param existingBranch @param newBranch @return @throws FatalException	This method will be used to update the account branch code while transferring the account from one branch to another.   This will internally call the {@link TaxManagementService#updateAccountBranch(DepositDetailsKey, String, String)} to update the branch . @param sessionContext @param partyID @param accountID @param existingBranch @param newBranch @return @throws FatalException
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	simulateTaxOnInterest	This method will be used to simulate tax on the amount if tax is applicable at compounding or capitalization for the given number of days. @param partyId @param arrangementNumber @param interestAmount @param dayCountConventionType @param numberOfDays @return	
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	simulateTaxOnInterest	This method will be used to simulate tax on the amount if tax is applicable at compounding or capitalization for the given number of days. @param partyId @param arrangementNumber @param interestAmount @param dayCountConventionType @param numberOfDays @return	This method inquires redemption details and will also calculate the maturity outstanding penalty amount, tax applicable and net interest payable on remaining deposit after redemption. @param redemptionRequestDTO {@link RedemptionRequestDTO} Object containing details required for premature redeem TD simulation @return {@link PrematureRedeemTDDTO} Object containing redemption details and maturity, interest details in case of premature part redeem simulation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	<p>This method acts as a bridge between TD module and Tax module&lt;br&gt;This method is used to register the Tax plan for Term Deposit with Tax module..&lt;br&gt;&lt;br&gt; @param deposit {@link Deposit} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.</p>	<p>The method de-register the deposit from Taxation module. @param deposit {@link Deposit} The term deposit for unregister</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#registerWithTax(com.ofss.fc.app.adapter.td.dto.TDTaxRegistrationDTO, com.ofss.fc.enumeration.taxation.ArrangementStatus)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	<p>This method acts as a bridge between TD module and Tax module&lt;br&gt;This method is used to register the Tax plan for Term Deposit with Tax module..&lt;br&gt;&lt;br&gt; @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.	This method is used for reversal of tax state stored for future reference. @param internalReferenceNumber Internal ref number @param originalTransactionReferenceNumber Transaction number used for reversal of tax journal, statistics etc. @see com.ofss.fc.app.adapter.td.ITDTaxationAdapter#reverseRedemption(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxation Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	<p>This method acts as a bridge between TD module and Tax module&lt;br&gt;This method is used to register the Tax plan for Term Deposit with Tax module..&lt;br&gt;&lt;br&gt; @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.</p>	<p>This method is used to payin deposit.&lt;br&gt; Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDTaxationAdapter	registerWithTax	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module..   @param deposit {@link DepositDTO} Object containing information about deposit for which interest plan needs to be registered. @param arrangementStatus {@link ArrangementStatus} Defines if the transaction is for registration/deregistration. if ArrangementStatus is Active then the transaction is for Registration. if ArrangementStatus is Close then the transaction is for deRegistration. @param isBatchMode : this variable is set to true if the method is called in batch mode.	This method acts as a bridge between TD module and Tax module This method is used to register the Tax plan for Term Deposit with Tax module.   @param deposit {@link DepositDTO} object containing information about deposit for which interest plan needs to be registered.
IPaymentsLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsLoanAdapter	clearFunds	@param accountID @param amount @param chequeNo @param moduleReferenceNo @return	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException
IPaymentsLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsLoanAdapter	updateUnclearFunds	@param accountTransactionDTO @return @throws FatalException	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doDailyInterestComputation	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed. @throws FatalException	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param LoanInterestEventRequestDTO @throws FatalException
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doDailyInterestComputation	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed. @throws	This method is called from {@link LoanInterestComputationService#executeInterestComputationAction(ApplicationContext, InterestComputationAction)}   during Batch processing to <strong>execute</strong>a interest computation action. This method will forward the call to  {@link LoanInterestAdapter#doDailyInterestComputation()}  @param applicationContext @param accountId @param effectiveDate @param loanInterestComputationLog



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doDailyInterestComputation	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed. @throws FatalException	This method is used to initiate penalty interest computation from due date. First it checks for unpaid arrears. If there are no unpaid arrears, it performs a regular interest computation otherwise it performs penalty computation from due date. @param applicationContext @param effectiveDate @param toleranceAmount @param loanAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doDailyInterestComputation	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed. @throws FatalException	@param accountId @param currentInterestDivision @param processDate @param includePrincipal @param includeArrears @param effectivedate @param toleranceAmount
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	registerInterestPlan	The method is responsible for registering an interest plan for the account in the interest module at the time of account opening.This method is referred from {@link DDAccountOpeningService#openDemandDepositAccount()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse } to register in interest module. InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product.	The method is responsible for registering an interest plan for the account in the interest module at the time of first disbursement.This method is referred from {@link DisbursementPostingService#post()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse } to register in interest module. InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product. {@link InterestPlanRegistrationDTO} in input will be populated as below: InterestPlanRegistrationDTO.interestRuleKeys: Map <String ,Date> :- This is key value pair of the IRD and the effective date. Effective date here is the account opening date i.e. postingDate To get the list of IRDs

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>{@link InterestPlanRegistrationDTO} in input will be populated as below:  InterestPlanRegistrationDTO.interestRuleKeys: Map &lt;String, Date&gt; :-  This is key value pair of the IRD and the effective date. Effective date here is the account opening date i.e. postingDate  To get the list of IRDs linked to product, call {@link IDDAProductAdapter#inquireProduct}  InterestPlanRegistrationDTO.accountId = accountId (input)  InterestPlanRegistrationDTO.depositNumber = ignore  InterestPlanRegistrationDTO.partyId = partyId (input)  InterestPlanRegistrationDTO.rate = ignore  balanceFromDate = postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interestDivision = ignore  FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext  domainCategory = DemandDeposit</p>	<p>linked to product, call {@link IDDAProductAdapter#inquireProduct}  InterestPlanRegistrationDTO.accountId = accountId (input)  InterestPlanRegistrationDTO.depositNumber = ignore  InterestPlanRegistrationDTO.partyId = partyId (input)  InterestPlanRegistrationDTO.rate = ignore  balanceFromDate = postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interestDivision = ignore  FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext  domainCategory = DemandDeposit  BankingFactContext.setProcessDate(Date processDate)  BankingFactContext.setAccountCurrency(String accountCurrency)  BankingFactContext.setAccountId(String accountId)  BankingFactContext.setPartyId(String partyId)  Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;;  set  InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set  InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param accountId @param processdate @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			BankingFactContext.setProcessDate(Date processDate) BankingFactContext.setAccountCurrency(String accountCurrency) BankingFactContext.setAccountId(String accountId) BankingFactContext.setPartyId(String partyId) Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map<String, Date>; set InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param accountId @param loanInterestEventRequestDTO	
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	registerInterestPlan	The method is responsible for registering an interest plan for the account in the interest module at the time of account opening. This method is referred from {@link DDAccountOpeningService#openDemandDepositAccount()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse } to register in interest module.	This method is called during <strong>first disbursement</strong> to register interest plan. This method checks the disbursement count of the account   be calling {@link LoanAccountService#fetchLoanAccount(accountId) } .    &nbsp;If the (LoanAccount.disbursement count) !=1   throw exception  else  call getLoanInterestAdapter().registerInterestPlan(accountId, processDate) to register. @param accountId This field represents Account Id @param applicationContext This field represents Application context required to derive process date. @param scheduleIdForPrincipalAmount TODO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product.</p> <p>{@link InterestPlanRegistrationDTO} in input will be populated as below:</p> <p>InterestPlanRegistrationDTO.interestRuleKeys: Map &lt;String ,Date&gt; :- This is key value pair of the IRD and the effective date. Effective date here is the account opening date i.e. postingDate To get the list of IRDs linked to product, call {@link IDDAProductAdapter#inquireProduct}</p> <p>InterestPlanRegistrationDTO.accountId = accountId (input)</p> <p>InterestPlanRegistrationDTO.depositNumber = ignore</p> <p>InterestPlanRegistrationDTO.partyId = partyId (input)</p> <p>InterestPlanRegistrationDTO.rate = ignore balanceFromDate = postingDate</p> <p>InterestPlanRegistrationDTO.balanceQualifier = ignore</p> <p>InterestPlanRegistrationDTO.balanceAmount = ignore</p> <p>InterestPlanRegistrationDTO.accountCurrency = accountCurrency</p> <p>InterestPlanRegistrationDTO.interestClass = ignore</p> <p>InterestPlanRegistrationDTO.interestDivision = ignore FactContext will be populated as below: create an object of</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			BankingFactContext(domainCategory) which extends FactContext domainCategory = DemandDeposit BankingFactContext.setProcessDate(Date processDate) BankingFactContext.setAccountCurrency(String accountCurrency) BankingFactContext.setAccountId(String accountId) BankingFactContext.setPartyId(String partyId) Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map<String, Date>; set InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param accountId @param loanInterestEventRequestDTO	
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	registerInterestPlan	The method is responsible for registering an interest plan for the account in the interest module at the time of account opening. This method is referred from {@link DDAccountOpeningService#openDemandDepositAccount()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO,	This method returns a map of LoanInterestComputationDetails between dates. This method checks the disbursement count of the account   by calling {@link LoanAccountService#fetchLoanAccount(accountId) } .    &nbsp;If the (LoanAccount.disbursement count) !=0   throw exception  else  call getLoanInterestAdapter().reverseInterestplanRegistration(accountId, processDate)} to unregister Interest plan. @param accountId @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>factContext: FactContext):  InterestPlanRegistrationResponse }  to register in interest module.  InterestPlanRegistrationResponse  contains a map of settlementDate for  each IRD linked to the product.  {@link InterestPlanRegistrationDTO}  in input will be populated as below:  InterestPlanRegistrationDTO.interestRuleKeys: Map &lt;String ,Date&gt; :-  This is key value pair of the IRD and  the effective date. Effective date  here is the account opening date i.e.  postingDate To get the list of IRDs  linked to product, call {@link  IDDAProductAdapter#inquireProduct}  InterestPlanRegistrationDTO.accountId = accountId (input)  InterestPlanRegistrationDTO.depositNumber = ignore  InterestPlanRegistrationDTO.partyId = partyId (input)  InterestPlanRegistrationDTO.rate = ignore  balanceFromDate =  postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interest</p>	<p>originalTransactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>tDivision = ignore FactContext will be populated as below: create an object of  BankingFactContext(domainCategory) which extends FactContext  domainCategory = DemandDeposit  BankingFactContext.setProcessDate(Date processDate)  BankingFactContext.setAccountCurrency(String accountCurrency)  BankingFactContext.setAccountId(String accountId)  BankingFactContext.setPartyId(String partyId) Return type  InterestPlanRegistrationResponse contains array of  InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;; set  InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set  InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param  accountId @param  loanInterestEventRequestDTO</p>	





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>ignore balanceFromDate = postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interestDivision = ignore  FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext  domainCategory = DemandDeposit  BankingFactContext.setProcessDate(Date processDate)  BankingFactContext.setAccountCurrency(String accountCurrency)  BankingFactContext.setAccountId(String accountId)  BankingFactContext.setPartyId(String partyId)  Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;; set InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set InterestPlanRegistrationResponseDTO.settlementDate in the second</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			argument of the map. @param accountId @param loanInterestEventRequestDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	registerInterestPlan	<p>The method is responsible for registering an interest plan for the account in the interest module at the time of account opening. This method is referred from <a href="#">{@link DDAccountOpeningService#openDemandDepositAccount()}</a> Call <a href="#">{@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse }</a> to register in interest module. InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product. <a href="#">{@link InterestPlanRegistrationDTO}</a> in input will be populated as below: InterestPlanRegistrationDTO.interestRuleKeys: Map &lt;String,Date&gt; :- This is key value pair of the IRD and the effective date. Effective date here is the account opening date i.e. postingDate To get the list of IRDs linked to product, call <a href="#">{@link IDDAProductAdapter#inquireProduct}</a> InterestPlanRegistrationDTO.accountId = accountId (input) InterestPlanRegistrationDTO.depositNumber = ignore InterestPlanRegistrationDTO.partyId = partyId (input) InterestPlanRegistrationDTO.rate =</p>	<p>This method is called from <a href="#">{@link PPFPostingService#post(ApplicationContext,EventData)}</a> to process interest for the broken period for Partial Payoff.   This method makes a call to <a href="#">{@link LoanInterestAdapter#processPartialPayOff(String, Date)}</a> to process the interest during Partial payoff posting. @param accountId @param date</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>ignore balanceFromDate = postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interestDivision = ignore  FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext  domainCategory = DemandDeposit  BankingFactContext.setProcessDate(Date processDate)  BankingFactContext.setAccountCurrency(String accountCurrency)  BankingFactContext.setAccountId(String accountId)  BankingFactContext.setPartyId(String partyId)  Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;; set InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set InterestPlanRegistrationResponseDTO.settlementDate in the second</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			argument of the map. @param accountId @param loanInterestEventRequestDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	renderCompoundingImpact	This method is used to forward the compounding request to<strong>Interest module</strong>.  First call to fecthInterestCompoundingRequest() to prepare<strong>InterestCompoundingRequest and</strong>  call {@link InterestCompoundingService#renderCompoundingImpact(InterestCompoundingRequest)}. @param accountId @param effectiveDate	This method is used to forward the compounding request to<strong>Interest module</strong>.  First call to fecthInterestCompoundingRequest() to prepare<strong>InterestCompoundingRequest and</strong>  call {@link InterestCompoundingService#renderCompoundingImpact(InterestCompoundingRequest)}. @param accountId @param effectiveDate
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	renderCompoundingImpact	This method is used to forward the compounding request to<strong>Interest module</strong>.  First call to fecthInterestCompoundingRequest() to prepare<strong>InterestCompoundingRequest and</strong>  call {@link InterestCompoundingService#renderCompoundingImpact(InterestCompoundingRequest)}. @param accountId @param effectiveDate	This method is called to initiate interest <strong>compounding</strong>. When the schedule is in <strong>MOR</strong> stage, the interest accrued for the account isn't charged until the stage end and is compounded . On the stage end date, the compounded interest is fetched and is either <strong>charged</strong>or <strong>capitalized</strong>as Principal. This method forwards the call to {@link LoanInterestAdapter.renderCompoundingImpact(String, Date)} @param applicationContext @param accountId
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reverseInterestPlanRegistration	This method is to be called during reversal of disbursement posting to un-register interest plan and stop interest computation.  @param accountId	This method is to be called during reversal of disbursement posting to un-register interest plan and stop interest computation.  @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reverseInterestPlanRegistration	This method is to be called during reversal of disbursement posting to un-register interest plan and stop interest computation.  @param accountId	This method returns a map of LoanInterestComputationDetails between dates. This method checks the disbursement count of the account   by calling {@link LoanAccountService#fetchLoanAccount(accountId) } .    &nbsp;If the (LoanAccount.disbursement count) !=0   throw exception  else  call getLoanInterestAdapter().reverseInterestplanRegistration(accountId, processDate)} to unregister Interest plan. @param accountId @param originalTransactionReferenceNumber



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedIntereset} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	<p>PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedIntereset} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date using &lt;br&gt; {@link ConstituentStatisticsForComputationRepository#load(Class, ConstituentStatisticsForComputationKey)} &lt;li&gt;Call the {@link InterestStateService#persistInterestState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputation)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareInterestCapitalizationRequest(payOffdate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInterestCapitalization(interestCapitalizationRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getPostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchargedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	<p>PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date using &lt;br&gt; {@link ConstituentStatisticsForComputationRepository#load(Class, ConstituentStatisticsForComputationKey)} &lt;li&gt;Call the {@link InterestStateService#persistInterestState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputation)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareInterestCapitalizationRequest(payOffdate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInterestCapitalization(interestCapitalizationRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getPostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchargedInterest} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<pre>&lt;pre&gt;The method will be used to inquire Partial pay off details. The method will validate if partial pay off is allowed for the account and the account is a valid, disbursed account. The method will return minimum transaction amount to perform PPF and account balances. Minimum partial pay off amount calculation on the basis of minimum number of installments for partial pay off maintained at product level. fc.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. fc.param MI,String,accountId, ,The parameter represents valid loan account for which partial payoff amount is to be inquired. fc.param MI,boolean,isUnbilledInterestToBeCharged, ,This flag tells system to whether Unbilled Interest will be charged as a part of partial pay off transaction. return {@link com.ofss.fc.app.loan.dto.LoanPartialPayOffInquiryResponse} throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {<a href="#">@link PartialPayOffResponse</a>} partialPayOffResponse; &lt;li&gt;If the {<a href="#">@link PartialPayOffRequest#applyInterest TillDate</a>]== false then&lt;br&gt; {<a href="#">@link PartialPayOffResponse#totalUnchargedInterest</a>} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {<a href="#">@link InterestClass</a>},{<a href="#">@link InterestDivision</a>}, fetch {<a href="#">@link InterestPlanItem</a>} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {<a href="#">@link ConstituentStatisticsForComputation</a>} for the given arrangement number, {<a href="#">@link InterestClass</a>},{<a href="#">@link InterestDivision</a>},{<a href="#">@link InterestConstituentType#REGULAR</a>} with respect to partial pay off date</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterest TillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method does partial payoff for a loan account through the provided CASA account. It provides partial payoff functionality through a simple API and as required by Dashboard. @param sessionContext @param CASAAccountNo @param loanAccountNo @param chequeNo @param chequeDate @param ppfAmount @return LoanPartialPayOffInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {<a href="#">@link PartialPayOffResponse</a>} partialPayOffResponse; &lt;li&gt;If the {<a href="#">@link PartialPayOffRequest#applyInterest TillDate</a>]== false then&lt;br&gt; {<a href="#">@link PartialPayOffResponse#totalUnchargedInterest</a>} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {<a href="#">@link InterestClass</a>},{<a href="#">@link InterestDivision</a>}, fetch {<a href="#">@link InterestPlanItem</a>} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {<a href="#">@link ConstituentStatisticsForComputation</a>} for the given arrangement number, {<a href="#">@link InterestClass</a>},{<a href="#">@link InterestDivision</a>},{<a href="#">@link InterestConstituentType#REGULAR</a>} with respect to partial pay off date</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method does partial payoff for a loan account through Cash. It provides partial payoff functionality through a simple API and as required by Dashboard. @param sessionContext @param loanAccountNo @param ppfAmount @param currencyDenominationDTO @return LoanPartialPayOffInquiryResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterest TillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>&lt;p&gt;This method provides facility to pay Loan partially by External Account.&lt;br&gt; It has been exclusively defined for DashBoard Application. &lt;/p&gt;         @param sessionContext @param externalAccountNumber @param loanAccountNumber @param ppfAmount @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterest TillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquirePartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method is called from {@link PPFPostingService#post(ApplicationContext,EventData)} to process interest for the broken period for Partial Payoff.&lt;br /&gt; &lt;br /&gt;This method makes a call to {@link LoanInterestAdapter#processPartialPayOff(String, Date)} to process the interest during Partial payoff posting. @param accountId @param date</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterest TillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	stopPreviousDivisionComputation	This method is called to stop interest computation for a particular division @param applicationContext @param accountId @param partyId @param loanBalanceInformationDTOMap @param InterestDivision	This method is used to initiate interest computation.     The caller supplies interest rate, computation bases and other details and this method forwards those value to fetchBalanceRecord().    And calls {@link InterestComputationService#doInterestComputation()}  @param accountId @param LoanInterestEventRequestDTO @throws FatalException
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	stopPreviousDivisionComputation	This method is called to stop interest computation for a particular division @param applicationContext @param accountId @param partyId @param loanBalanceInformationDTOMap @param InterestDivision	@param accountId @param currentInterestDivision @param processDate @param includePrincipal @param includeArrears @param effectiveDate @param toleranceAmount
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		This method will in turn call doInterestCapitalization() in InterestCapitalizationService . This is called during charging to capitalize accrued interest and to return the interest amount to the callee.  First InterestCapitalizationRequest is prepared and then it is supplied to Interest module to initiate capitalization process. The response is converted into InterestBalancemap by calling prepareInterestBalanceMap().  @param accountId @param dueDate @param isAgreementBeingClosed @param interestDivision @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		This method does regular interest charging (Normal and Rollover interest). It also adjusts accrual for the difference in total amount accrued and total amount charged. 1.Raise the arrear for regular interest amount to be billed. @param applicationContext @param accountId @param productCode @param accrualStatusType @param accountCurrencyCode @param dueDate @param eventId @param actualInterest @param interestBalance @param amortizedRolloverInterest @param rolloverInterest @param interestRuleCode @param interestType @param installmentNumber @param txnEventId @param isAccrualToBeAdjusted @param expectedInterest
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		The method will capitalise Interest and fee arrears which were not raised during hardship relief period. @param context @param loanAccount @param reliefEffectiveDate @param reliefExpiryDate @throws FatalException
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		This method will capitalise interest arrears which were not raised during hardship relief period. @param context @param loanAccount
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		This method is called for Interest charging to be occurred during charging. Interest module is notified and the respective balances are fetched for all interest division. This method fetches the latest LoanInterestComputationLog by calling {@link LoanInterestComputationLog#fetchCurrentInterestComputationDetails(AccountId)} then it checks for current Interest division by calling {@link LoanInterestComputationLog.getInterestDivision()}; If( Interest Division ==Regular or Penalty) call{@link LoanInterestAdapter#doInterestCapitalization(AccountId, dueDate, false, Regular)} @param accountId @param txnEventId @param effectiveDate
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doInterestCapitalization		This method is called for Interest charging to be occurred during charging. Interest module is notified and the respective balances are fetched for all interest division. This method fetches the latest LoanInterestComputationLog by calling {@link LoanInterestComputationLog#fetchCurrentInterestComputationDetails(AccountId)} then it checks for current Interest division by calling {@link LoanInterestComputationLog.getInterestDivision()}; If( Interest Division ==Regular or Penalty) call{@link LoanInterestAdapter#doInterestCapitalization(AccountId, dueDate, false, Regular)} @param accountId @param effectiveDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	processPartialPayOff	This method will called by the consumers of the interest module to process the partial pay off request.  There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.  This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle.   If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.  The below steps are applicable if the applyInterestTillDate flag is set as true This method would perform the interest charging as of the partial pay off date provided in the input.   Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off.   The same transaction reference number would be used to raise the interest related accounting entries.   So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest	This method will called by the consumers of the interest module to process the partial pay off request.  There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.  This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle.   If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.  The below steps are applicable if the applyInterestTillDate flag is set as true This method would perform the interest charging as of the partial pay off date provided in the input.   Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off.   The same transaction reference number would be used to raise the interest related accounting entries.   So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest accounting entries.   The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries.   The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.    Algorithm : <ol> <li>Initialize {@link PartialPayOffResponse} partialPayOffResponse; <li>If the {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	<p>PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date using &lt;br&gt; {@link ConstituentStatisticsForComputationRepository#load(Class, ConstituentStatisticsForComputationKey)} &lt;li&gt;Call the {@link InterestStateService#persistInterestState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputation)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareInterestCapitalizationRequest(payOffdate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInterestCapitalization(interestCapitalizationRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getPostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchargedInterest} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	processPartialPayOff	<p>This method will called by the consumers of the interest module to process the partial pay off request.&lt;br&gt; There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.&lt;br&gt; This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle. &lt;br&gt; If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.&lt;br&gt; The below steps are applicable if the applyInterestTillDate flag is set as true&lt;br&gt;This method would perform the interest charging as of the partial pay off date provided in the input. &lt;br&gt; Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off. &lt;br&gt; The same transaction reference number would be used to raise the interest related accounting entries. &lt;br&gt; So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest</p>	<p>This method is called from {@link PPFPostingService#post(ApplicationContext,EventData)} to process interest for the broken period for Partial Payoff.&lt;br /&gt; &lt;br /&gt;This method makes a call to {@link LoanInterestAdapter#processPartialPayOff(String, Date)} to process the interest during Partial payoff posting. @param accountId @param date</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>accounting entries. &lt;br&gt; The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries. &lt;br&gt; The current state of the interest plan item would be preserved, so that during the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterest TillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> using &lt;br&gt; {@link ConstituentStatisticsForComputatio nRepository#load(Class, ConstituentStatisticsForComputatio nKey)} &lt;li&gt;Call the {@link InterestStateService#persistInteres tate(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputatio n)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareIn terestCapitalizationRequest(payOffd ate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInte restCapitalization(interestCapitaliza tionRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getP ostTaxInterestAmount()} &lt;li&gt;Set {@link PartialPayOffResponse#totalUnchar gedIntereset} = accruedAmount; &lt;li&gt;return the response to the caller. &lt;/ol&gt; &lt;/ol&gt; @param sessionContext @param partialPayOffRequest </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reversePartialPayOff	<p>In certain cases the customer may wishes to reverse a partial pay off made earlier.&lt;br&gt; In such case, the interest buckets would be restored to its original state as that of pre-partial pay off state. &lt;br&gt; All the accounting entries that were passed during the partial pay off would be reversed.&lt;br&gt; But the reversal of the accounting entries would be triggered by the loan module.&lt;br&gt; The logs which were generated during the partial pay off would be removed.&lt;br&gt; If there was interest adjustment done after the partial pay off, even after the reversal of the event would still reflect in the interest buckets. &lt;br&gt; Algorithm :</p> <p>&lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;{@link Adjustment} adjustedAccrued = {@link AccrualStatistics#getAdjustment()} &lt;li&gt;{@link Adjustment} adjustedCapitalised = {@link CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call {@link InterestStateService#fetchInterestState(transactionReferenceNumber)} to get {@link InterestState} &lt;li&gt;{@link InterestPlanItem} fetchedinterestPlanItem = {@link InterestState#getInterestPlanItem()} &lt;li&gt;{@link ConstituentStatisticsForComputation} constituentStatisticsForComputation = {@link InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link InterestRuleKey} interestRuleKey = {@link InterestPlanItem#getInterestRuleKey()} &lt;li&gt;Call {@link ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set {@link AccrualStatistics#setAdjustment()} of fetchedinterestPlanItem = {@link</p>	<p>In certain cases the customer may wishes to reverse a partial pay off made earlier.&lt;br&gt; In such case, the interest buckets would be restored to its original state as that of pre-partial pay off state. &lt;br&gt; All the accounting entries that were passed during the partial pay off would be reversed.&lt;br&gt; But the reversal of the accounting entries would be triggered by the loan module.&lt;br&gt; The logs which were generated during the partial pay off would be removed.&lt;br&gt; If there was interest adjustment done after the partial pay off, even after the reversal of the event would still reflect in the interest buckets. &lt;br&gt; Algorithm :</p> <p>&lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;{@link Adjustment} adjustedAccrued = {@link AccrualStatistics#getAdjustment()} &lt;li&gt;{@link Adjustment} adjustedCapitalised = {@link CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call {@link InterestStateService#fetchInterestState(transactionReferenceNumber)} to get {@link InterestState} &lt;li&gt;{@link InterestPlanItem} fetchedinterestPlanItem = {@link InterestState#getInterestPlanItem()} &lt;li&gt;{@link ConstituentStatisticsForComputation} constituentStatisticsForComputation = {@link InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link InterestRuleKey} interestRuleKey = {@link InterestPlanItem#getInterestRuleKey()} &lt;li&gt;Call {@link ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set {@link AccrualStatistics#setAdjustment()} of fetchedinterestPlanItem = {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestState#getInterestPlanItem()  &lt;li&gt;{@link  ConstituentStatisticsForComputation}  constituentStatisticsForComputation  = {@link  InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link  InterestRuleKey} interestRuleKey =  {@link  InterestPlanItem#getInterestRuleKey()  }&lt;li&gt;Call {@link  ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set  {@link  AccrualStatistics#setAdjustment()}  of fetchedinterestPlanItem = {@link  AccrualStatistics#getAdjustment()}  &lt;li&gt;Set {@link  CapitalizationStatistics#setAdjustment()}  of fetchedinterestPlanItem =  {@link  CapitalizationStatistics#getAdjustment()}  }&lt;li&gt;Call {@link  InterestPlanItemRepository#update(fetchedinterestPlanItem)} &lt;li&gt;Call  {@link  EventBasedLogItemRepository#removeForRange(arrangementNumber,  InterestRuleKey, payOffDate,  payOffDate)}&lt;br&gt; This will be done  for both accrual and capitalization  logs @param orgReferenceNumber</p>	<p>AccrualStatistics#getAdjustment() &lt;li&gt;Set {@link  CapitalizationStatistics#setAdjustment()}  of  fetchedinterestPlanItem = {@link  CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call  {@link  InterestPlanItemRepository#update(fetchedinterestPlanItem)} &lt;li&gt;Call  {@link  EventBasedLogItemRepository#removeForRange(arrangementNumber,  InterestRuleKey, payOffDate,  payOffDate)}&lt;br&gt; This will be done  for both accrual  and capitalization logs @param orgReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reversePartialPayOff	<p>In certain cases the customer may wishes to reverse a partial pay off made earlier.&lt;br&gt; In such case, the interest buckets would be restored to its original state as that of pre-partial pay off state. &lt;br&gt; All the accounting entries that were passed during the partial pay off would be reversed.&lt;br&gt; But the reversal of the accounting entries would be triggered by the loan module.&lt;br&gt; The logs which were generated during the partial pay off would be removed.&lt;br&gt; If there was interest adjustment done after the partial pay off, even after the reversal of the event would still reflect in the interest buckets. &lt;br&gt; Algorithm :</p> <pre> &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;{@link Adjustment} adjustedAccrued = {@link AccrualStatistics#getAdjustment()} &lt;li&gt;{@link Adjustment} adjustedCapitalised = {@link CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call {@link InterestStateService#fetchInterestState(transactionReferenceNumber)} to get {@link InterestState} &lt;li&gt;{@link InterestPlanItem} fetchedinterestPlanItem = {@link </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestState#getInterestPlanItem()  &lt;li&gt;{@link  ConstituentStatisticsForComputation}  constituentStatisticsForComputation  = {@link  InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link  InterestRuleKey} interestRuleKey =  {@link  InterestPlanItem#getInterestRuleKey()  }() &lt;li&gt;Call {@link  ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set  {@link  AccrualStatistics#setAdjustment()}  of fetchedinterestPlanItem = {@link  AccrualStatistics#getAdjustment()}  &lt;li&gt;Set {@link  CapitalizationStatistics#setAdjustment()}  of fetchedinterestPlanItem =  {@link  CapitalizationStatistics#getAdjustment()}  }() &lt;li&gt;Call {@link  InterestPlanItemRepository#update(fetchedinterestPlanItem)} &lt;li&gt;Call  {@link  EventBasedLogItemRepository#removeForRange(arrangementNumber,  InterestRuleKey, payOffDate,  payOffDate)}&lt;br&gt; This will be done  for both accrual and capitalization  logs @param orgReferenceNumber</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reversePartialPayOff	<p>In certain cases the customer may wishes to reverse a partial pay off made earlier.&lt;br&gt; In such case, the interest buckets would be restored to its original state as that of pre-partial pay off state. &lt;br&gt; All the accounting entries that were passed during the partial pay off would be reversed.&lt;br&gt; But the reversal of the accounting entries would be triggered by the loan module.&lt;br&gt; The logs which were generated during the partial pay off would be removed.&lt;br&gt; If there was interest adjustment done after the partial pay off, even after the reversal of the event would still reflect in the interest buckets. &lt;br&gt; Algorithm :</p> <pre> &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;{@link Adjustment} adjustedAccrued = {@link AccrualStatistics#getAdjustment()} &lt;li&gt;{@link Adjustment} adjustedCapitalised = {@link CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call {@link InterestStateService#fetchInterestState(transactionReferenceNumber)} to get {@link InterestState} &lt;li&gt;{@link InterestPlanItem} fetchedinterestPlanItem = {@link </pre>	<p>During &lt;strong&gt;Partial Payoff reversal, &lt;/strong&gt;this method is called to reverse the broken period interest capitalized during normal transaction. &lt;br /&gt;This method calls {@link LoanInterestAdapter#reversePartialPayoff(String, String)}&amp;nbsp; to reverse . @param accountId @param originalTransactionReference</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestState#getInterestPlanItem()  &lt;li&gt;{@link  ConstituentStatisticsForComputation}  constituentStatisticsForComputation  = {@link  InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link  InterestRuleKey} interestRuleKey =  {@link  InterestPlanItem#getInterestRuleKey()  }() &lt;li&gt;Call {@link  ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set  {@link  AccrualStatistics#setAdjustment()}  of fetchedinterestPlanItem = {@link  AccrualStatistics#getAdjustment()}  &lt;li&gt;Set {@link  CapitalizationStatistics#setAdjustment()}  of fetchedinterestPlanItem =  {@link  CapitalizationStatistics#getAdjustment()}  }() &lt;li&gt;Call {@link  InterestPlanItemRepository#update(fetchedinterestPlanItem)} &lt;li&gt;Call  {@link  EventBasedLogItemRepository#removeForRange(arrangementNumber,  InterestRuleKey, payOffDate,  payOffDate)}&lt;br&gt; This will be done  for both accrual and capitalization  logs @param orgReferenceNumber</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reversePartialPayOff	<p>In certain cases the customer may wishes to reverse a partial pay off made earlier.&lt;br&gt; In such case, the interest buckets would be restored to its original state as that of pre-partial pay off state. &lt;br&gt; All the accounting entries that were passed during the partial pay off would be reversed.&lt;br&gt; But the reversal of the accounting entries would be triggered by the loan module.&lt;br&gt; The logs which were generated during the partial pay off would be removed.&lt;br&gt; If there was interest adjustment done after the partial pay off, even after the reversal of the event would still reflect in the interest buckets. &lt;br&gt; Algorithm :</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;{@link Adjustment} adjustedAccrued = {@link AccrualStatistics#getAdjustment()} &lt;li&gt;{@link Adjustment} adjustedCapitalised = {@link CapitalizationStatistics#getAdjustment()} &lt;li&gt;Call {@link InterestStateService#fetchInterestState(transactionReferenceNumber)} to get {@link InterestState} &lt;li&gt;{@link InterestPlanItem} fetchedinterestPlanItem = {@link</li> </ul>	<p>This method is defined for posting reversal of Partial PayOff. @param applicationContext @param eventData @param originatingTransactionReferenceNumber</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestState#getInterestPlanItem()  &lt;li&gt;{@link  ConstituentStatisticsForComputation}  constituentStatisticsForComputation  = {@link  InterestState#getConstituentStatisticsForComputation()} &lt;li&gt;{@link  InterestRuleKey} interestRuleKey =  {@link  InterestPlanItem#getInterestRuleKey()  }() &lt;li&gt;Call {@link  ConstituentStatisticsForComputationRepository#update(constituentStatisticsForComputation)} &lt;li&gt;Set  {@link  AccrualStatistics#setAdjustment()}  of fetchedinterestPlanItem = {@link  AccrualStatistics#getAdjustment()}  &lt;li&gt;Set {@link  CapitalizationStatistics#setAdjustment()}  of fetchedinterestPlanItem =  {@link  CapitalizationStatistics#getAdjustment()}  }() &lt;li&gt;Call {@link  InterestPlanItemRepository#update(fetchedinterestPlanItem)} &lt;li&gt;Call  {@link  EventBasedLogItemRepository#removeForRange(arrangementNumber,  InterestRuleKey, payOffDate,  payOffDate)}&lt;br&gt; This will be done  for both accrual and capitalization  logs @param orgReferenceNumber</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	processAccountClosure		This method is called to process account closure during EFS.
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	processAccountClosure		This method is called during normal account closure.
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	processAccountClosure		This method is called from <a href="#">{@link PPFPostingService#post(ApplicationContext,EventData)}</a> to process interest for the broken period for Partial Payoff.  This method makes a call to <a href="#">{@link ILoanInterestAdapter#processPartialPayOff(String,Date)}</a> to process the interest during Partial payoff posting. @param accountId @param date
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	unRegisterInterestPlan		This method is called during normal account closure.
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	unRegisterInterestPlan		This method returns a map of LoanInterestComputationDetails between dates. This method checks the disbursement count of the account  by calling <a href="#">{@link LoanAccountService#fetchLoanAccount(accountId)}</a>  &nbsp;If the (LoanAccount.disbursement count) !=0  throw exception  else  call <a href="#">getLoanInterestAdapter().reverseInterestplanRegistration(accountId, processDate)}</a> to unregister Interest plan. @param accountId @param originalTransactionReferenceNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	unRegisterInterestPlan		This method is used to unregister Interest plan during account closure @param applicationContext @param accountId
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireInterest		This method will called by the consumers of the interest module to process the partial pay off request.  There is an attribute defined at the product level which defines whether the pending interest needs to be charged during Partial Payoff.  This attribute is termed as applyInterestTillDate. This attribute will decide the whether the pending interest will be charged as a part of Partial Payoff processing or it will be charged in the next charging cycle.   If the applyInterestTillDate flag is set as false in the input , then pending interest won't be charged hence zero money response would be sent to the caller.  The below steps are applicable if the applyInterestTillDate flag is set as true This method would perform the interest charging as of the partial pay off date provided in the input.   Here the transaction reference number is taken as input from the loan module, which is generated during the partial pay off.   The same transaction reference number would be used to raise the interest related accounting entries.   So when there is reversal for the partial pay off, the same transaction reference number could be used to reverse the interest accounting entries.   The reversal of the accounting entries would be triggered by the loan module. Doing so the same transaction reference number would be used to reverse all the accounting entries. It will reverse both loan and interest module accounting entries.   The current state of the interest plan item would be preserved, so that during

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the reversal the interest plan item could be brought to its original state as of pre-partial pay off.&lt;br&gt; &lt;br&gt; Algorithm : &lt;ol&gt; &lt;li&gt;Initialize {@link PartialPayOffResponse} partialPayOffResponse; &lt;li&gt;If the {@link PartialPayOffRequest#applyInterestTillDate}== false then&lt;br&gt; {@link PartialPayOffResponse#totalUnchargedInterest} = 0; &lt;li&gt;return the response to the caller. &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;For given arrangement number, {@link InterestClass},{@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to partial pay off date. &lt;li&gt;Fetch the {@link ConstituentStatisticsForComputation} for the given arrangement number, {@link InterestClass},{@link InterestDivision} ,{@link InterestConstituentType#REGULAR} with respect to partial pay off date using &lt;br&gt; {@link ConstituentStatisticsForComputationRepository#load(Class, ConstituentStatisticsForComputationKey)} &lt;li&gt;Call the {@link InterestStateService#persistInterestState(String, InterestPlanItem, Date, isInterestActionStateRequired, ConstituentStatisticsForComputation)} &lt;li&gt;Prepare {@link InterestCapitalizationRequest} interestCapitalizationRequest = {@link InterestProcessingHelper#prepareInterestCapitalizationRequest(payOffdate, payOffdate, InterestPlanItem, isBackdating, transactionReferenceNumber)} &lt;li&gt;Call {@link InterestCapitalizationService#doInterestCapitalization(interestCapitalizationRequest)} &lt;li&gt;Money accruedAmount = {@link InterestCapitalizationResponse#getPostTaxInterestAmount()} &lt;li&gt;Set {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				PartialPayOffResponse#totalUnchargedInterest) = accruedAmount; <li>return the response to the caller. </ol> </ol> @param sessionContext @param partialPayOffRequest
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireInterest		This method will in turn call fetchAllInterestPlanItemDetails() in InterestInquiryService .This method is call to fetch all the calculated interest Amount (both accrued and charged). @param accountId @param interestDivision @param interestDivision @throws FatalException @returns InterestPlanItemsResponse
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireInterest		
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireInterest		This method will be used to inquire about the interest benefit period @param accountId @param effectiveDate @return interestCyclePeriodResponse
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireInterest		This method will decorate the ArrearByType[] passed as input to this function with the below arrear types: 1) Normal Regular Interest Accrued 2) Suspended Regular Interest Accrued 3) Normal PMI Interest Accrued 4) Suspended PMI Interest Accrued 5) Normal Penalty Interest Accrued 6) Suspended Penalty Interest Accrued 7) Outstanding Principal @param applicationContext @param accountId @param arrearsByType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	startSuspendedAccrual	This method calls the startSuspendedAccrual() of InterestOperationService. Following parameters are defaulted byThis method: <li>interestClass is set as InterestClass.Debit</li> <li>effectiveDate is sent as null</li> <li> currentDate is sent as process Date</li> @param appContext @param accountId @param isClawback	This method calls the startSuspendedAccrual() of InterestOperationService. Following parameters are defaulted byThis method: <li>interestClass is set as InterestClass.Debit</li> <li>effectiveDate is sent as null</li> <li> currentDate is sent as process Date</li> @param appContext @param accountId @param isClawback
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	startSuspendedAccrual	This method calls the startSuspendedAccrual() of InterestOperationService. Following parameters are defaulted byThis method: <li>interestClass is set as InterestClass.Debit</li> <li>effectiveDate is sent as null</li> <li> currentDate is sent as process Date</li> @param appContext @param accountId @param isClawback	This method is called to set the account's Accrual status and CRR Code. This method should be called when during classification batch the Accrual Status is being changed for an account from Normal to Suspended. It performs following processing: 1) It updates the loan account's CRR code and Accrual Status. 2) It loops through the ArrearByTypeDTO[] array and for each of the interest type of arrear and calls LoanInterestAdapter.startSuspendedAccrual(). 3) It calls the ArrearService.classifyAccountAsSuspended() method @param applicationContext @param accountId @param arrear @param clawbackPrefDTO @param crrCode
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	stopSuspendedAccrual	This method calls the stopSuspendedAccrual() of InterestOperationService. Following parameters are defaulted byThis method: @param appContext @param accountId	This method calls the stopSuspendedAccrual() of InterestOperationService. @param appContext @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	stopSuspendedAccrual	This method calls the stopSuspendedAccrual() of InterestOperationService. Following parameters are defaulted by This method: @param appContext @param accountId	This method is called to set the accounts accrual status and CRR Code. This method should be called when during classification batch the accrual status is being changed for an account from Suspended to Normal. It performs following processing: 1) It updates the loan account's CRR code and Accrual Status. 2) It calls the LoanInterestAdapter.stopSuspendedAccrual() method. 3) It calls the ArrearService.classifyAccountAsNormal() method @param applicationContext @param accountId @param crrCode
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	writeOffAccount	This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId	This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	writeOffAccount	This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId	This method is called to fully writeoff an account. @param applicationContext @param eventId @param accountId @param reason1 @param reason2 @param provisionDrawdownAmount
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	intimateInterestForAccountTransfer	Method for Account Branch Transfer @param reviseInterestForBrokenPeriod @param balanceRecordsSet @throws FatalException	Method intimateInterest will be used to intimate the interests while transferring the account from one branch to other branch. @param accountId @param casaDepositNumber @param newBranchId @param transferDate
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	freezeInterestCharging	This method is called to freeze interest charging @param processDate @param accountId @param reliefEffectiveDate @param reliefExpiryDate @param isModification	This method is called to notify interest for freezing interest charging. @param applicationContext @param accountId @param reliefEffectiveDate @param reliefExpiryDate @param isModification

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireUnbilledInterest	This method is used to inquire Adhoc interest @param accountId @param isClosure @return	This method returns the interest till date taking into consideration adjustments for FPI installments. @param account @param balances @param stage @param effectiveDate @return Money
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireUnbilledInterest	This method is used to inquire Adhoc interest @param accountId @param isClosure @return	This method is used to inquire Accrued interest Amount. Interest Division is optional input. If Interest Division is null interest Amount Accrued for Post maturity, Regular and Penalty are fetched . Else Interest amount for given Interest Division is fetched. This method makes a call to {@link ILoanInterestAdapter#inquireUnbilledInterest(String, Date, boolean)} to fetch the InterestPlanItemsResponse. From InterestPlanItemsResponse interest Amount Accrued is fetched and stored in the map as InterestDivision,Money @param appContext @param accountId @param interestDivision @return Map<InterestDivision, Money>
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	fetchInterestPlanItemDetails	This method will be used to fetch the details related to {@link InterestPlanItem} from Interest module.  @param accountId @param effectiveDate @return InterestPlanItemDetailsResponse @throws FatalException	This method will be used to fetch the details related to {@link InterestPlanItem} from Interest module.  @param accountId @param effectiveDate @return InterestPlanItemDetailsResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	cancelInterestFreeze	@param processDate @param accountId	This is the private method which gives a call to the interest module for freezing the interest charging. It prepares the {@link InterestHardshipReliefRequest} with the inputs coming in and sends the request to the {@link InterestFreezeService#performInterestFreeze(InterestHardshipReliefRequest)} to perform interest freeze and update the {@link InterestPlanItem}. @param processDate @param accountId @param reliefEffectiveDate @param reliefExpiryDate @param isModification @param interestDivision
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	cancelInterestFreeze	@param processDate @param accountId	@param processDate @param accountId @param interestDivision TODO
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	cancelInterestFreeze	@param processDate @param accountId	@param applicationContext @param accountId
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	calculateInterestBenefit	This method is used to calculate the interest benefit @param interestBenefitRequest @return	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	getInterestPeriodForCurrentAndNextCycle	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @param interestClass @return	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @return interestCyclePeriodResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interestCyclePeriodResponse	
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	interestBenefitInquiryResponse	This method will be used to inquire about the interest benefit period @param accountId @param effectiveDate @param interestClass @param registrationDate TODO @return interestCyclePeriodResponse	This method will be used to inquire about the interest benefit period @param accountId @param effectiveDate @return interestCyclePeriodResponse
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	interestBenefitInquiryResponse	This method will be used to inquire about the interest benefit period @param accountId @param effectiveDate @param interestClass @param registrationDate TODO @return interestCyclePeriodResponse	This method will be used to log interest benefit action on account having offset facility.  This action will be executed in batch and will be used to calculate the interest benefit enjoyed by   account due to having offset facility. @param loanAccount
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	performAccrualForEFSChequeDeposit	This method would invoke {@link InterestOperationService#performAccrualForEFSChequeDeposit} which would stop accrual of interest for the provided accountId. @param applicationContext @param accountId @param isArrangementBeingClosed	This method will post the normal transaction that is the transaction is not  value dated transaction(back-dated Transaction), therefore this method will do the appropriation also here only. @param applicationContext @param eventData @throws FatalException
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	performAccrualForEFSChequeDeposit	This method would invoke {@link InterestOperationService#performAccrualForEFSChequeDeposit} which would stop accrual of interest for the provided accountId. @param applicationContext @param accountId @param isArrangementBeingClosed	This method would invoke {@link InterestOperationService#performAccrualForEFSChequeDeposit} which would stop accrual of interest for the provided loan accountId as EFS is request through cheque. @param applicationContext @param accountId @param isArrangementBeingClosed

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	performInterestProcessingForEFSChequeClearing	This method would invoke {@link InterestOperationService#performInterestProcessingForEFSChequeClearing} which would @param applicationContext @param accountId @param isArrangementBeingClosed @param isChequeDishonoured	This method would invoke {@link InterestOperationService#performInterestProcessingForEFSChequeClearing} which would @param applicationContext @param accountId @param isArrangementBeingClosed @param isChequeDishonoured @return Map<InterestDivision, HashMap<AccrualStatus, Money>>
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	performInterestProcessingForEFSChequeClearing	This method would invoke {@link InterestOperationService#performInterestProcessingForEFSChequeClearing} which would @param applicationContext @param accountId @param isArrangementBeingClosed @param isChequeDishonoured	This method calls the interest module to get the regular and penalty interest accrued till the day before EFS posting date, and raises arrears for the same. @param applicationContext @param accountId @param accountCurrencyCode @param dueDate @param eventId @param loanInterestType @param installmentNumber @param isArrangementBeingClosed @param isChequeDishonoured
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	performInterestProcessingForEFSChequeClearing	This method would invoke {@link InterestOperationService#performInterestProcessingForEFSChequeClearing} which would @param applicationContext @param accountId @param isArrangementBeingClosed @param isChequeDishonoured	This method is called for book balance in case if there exist a dishonour for the cheque clearance. @param transactionDetailsDTO @param applicationContext @param amountInAccountCurrency @throws FatalException
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	This method is used to update next capitalization date in case if the interest cap/charge frequency of the schedule is getting updated This is required because Interest module accrues interest till next cap/charge date. If the frequency of the interest charging/cap changes we need to notify interest regarding the same, so the accrual will be in sync with the interest charge/cap frequency of the schedule

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	This method posts the EFS posting rule for a particular loan account @param applicationContext @param action
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	@param accountId @param existingAccountState @param newAccountState @param loanActionType
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	This method saves schedule on restructure @param applicationContext @param accountId @param scheduleTypeCode @param restructureDate @param reasonCode @param reasonDescription @param schedule @param blownupSchedule
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	This method is used to fetch the economic cost and fixed rate details for an accountId and eventId</br> @param context @param accountId @param feeCollectionDTO @param eventId @param ppfAmount
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	updateNextInterestCapDate	@param accountId @param nextChargingDate @param isNewCharging	This method posts the PPF posting rule for a particular loan account @param applicationContext @param eventData

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	logOffsetBenefitActions	@param accountNumber	This method is used to log the offset benefit actions explicitly when the event date is equal to the charging date due to which the previous offset actions get removed but are not logged as a part of {@link ILoanInterestAdapter#updateNextInterestCapDate(String, Date, boolean)} since the dueDate sent as input is event date minus one day which is before the charging date. This has been added only to the cases when the charging falls on the same day as restructure. The offset benefit actions need to be logged only for {@link InterestClass#DEBIT} and {@link InterestDivision#REGULAR}. @param accountNumber
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	logOffsetBenefitActions	@param accountNumber	This method saves schedule on restructure @param applicationContext @param accountId @param scheduleTypeCode @param restructureDate @param reasonCode @param reasonDescription @param schedule @param blowupSchedule
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doBackdatedInterestComputation	@param accountId @param period @param computePenalty	@param appContext @param accountId @param processingPeriod @param nextCapDueDate
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doBackdatedInterestComputation	@param accountId @param period @param computePenalty	
ILoanInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	doBackdatedInterestComputation	@param accountId @param period @param computePenalty	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireBackdatedAccruedInterest	@param accountId @param fromDate @param toDate @param isHardshipRelief @return	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireBackdatedAccruedInterest	@param accountId @param fromDate @param toDate @param isHardshipRelief @return	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireBackdatedAccruedInterest	@param accountId @param fromDate @param toDate @param isHardshipRelief @return	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	inquireBackdatedAccruedInterestForIPI	@param accountId @param interestClass @param interestDivision @param fromDate @param toDate @param isHardshipRelief @return	
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	calculateScheduleInterest	@param applicationContext @param loansInterestSimulationRequest @return InterestSimulationResponse	This method calculates the schedule interest for the simulation request. @param applicationContext @param loansInterestSimulationRequest @return InterestSimulationResponse
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	calculateScheduleInterest	@param applicationContext @param loansInterestSimulationRequest @return InterestSimulationResponse	<p>This method is responsible for computing the interest amount for bridge loan.   </p> @param sessionContext @param loanAccountConfigurationDTO @return interestAmountForBridge @throws FatalException
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	calculateScheduleInterest	@param applicationContext @param loansInterestSimulationRequest @return InterestSimulationResponse	This method will fetch interest amount for given date range, for current loan outstanding principal balance and applicable rate as on startDate @param applicationContext @param accountId @param startDate @param endDate @return @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reverseInterestAccruedForLoanRescission	This method would call the <a href="#">InterestOperationService#reverseInterestAccruedForLoanRescission</a> to reverse the accrued amount in case of Loan Rescission. @param accountNumber @param interestClass @param interestDivision @param transactionReferenceNumber @throws FatalException	This method calls the <a href="#">InterestOperationService#reverseInterestAccruedForLoanRescission(String, InterestClass, InterestDivision, String)</a> to reverse the accrued interest for the current cycle during loan rescission.
ILoanInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInterestAdapter	reverseInterestAccruedForLoanRescission	This method would call the <a href="#">InterestOperationService#reverseInterestAccruedForLoanRescission</a> to reverse the accrued amount in case of Loan Rescission. @param accountNumber @param interestClass @param interestDivision @param transactionReferenceNumber @throws FatalException	This method is called from <a href="#">PPFPostingService#post(ApplicationContext, EventData)</a> to process interest for the broken period for Partial Payoff.  This method makes a call to <a href="#">LoanInterestAdapter#processPartialPayOff(String, Date)</a> to process the interest during Partial payoff posting. @param accountId @param date
IPaymentsUtilityBillAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsUtilityBillAdapter	markBillAsFailed	This method will mark the bill payment as failure. It may return accounting details for reversing the credit given to the biller. @param context @param billerId @param transactionAmount @param eventId @param billerPreferenceId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsUtilityBillAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsUtilityBillAdapter	remitFundsToBillerOnValueDateClearing	This method will proceed with the released funds and remit it to the biller. It will only deal with the credit perspective of the bill payments, and return its accounting details. @param billerId @param transactionAmount @param listAccountingEventAmountTagDetailsDTO @param eventId @param moduleReferenceNo @return @throws FatalException	
IPaymentsUtilityBillAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsUtilityBillAdapter	remitFundsToBillerOnValueDateClearing	This method will proceed with the released funds and remit it to the biller. It will only deal with the credit perspective of the bill payments, and return its accounting details. @param billerId @param transactionAmount @param listAccountingEventAmountTagDetailsDTO @param eventId @param moduleReferenceNo @return @throws FatalException	
ICollateralPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralPartyAdapter	fetchPartyDetails	This method returns party details for a given party id. @param partyId @return	
ICollateralPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralPartyAdapter	fetchPartyDetails	This method returns party details for a given party id. @param partyId @return	This method check is party deceased. @param partyId @return Boolean



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralTitleSearchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titesearch\ICollateralTitleSearchProcessAdapter	callTitleSearchClientProcessForCollateralUpdate		This method will be invoked when , the <b>&lt;b&gt;cancellation&lt;/b&gt;</b> request has been <b>&lt;b&gt;confirmed&lt;/b&gt;</b> from Valex. This will result in the request status being set to <u>cancelled&lt;/u&gt;. @fcb.param SessionContext,sessionContext, Holds the basic information about the user of the system. @fcb.param CollateralTitleSearchCancelConfirmationDTO, collateralTitleSearchCancelConfirmationDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance <b>&lt;br&gt;</b> of TransactionStatus contains the information whether the transaction is success\failure. <b>&lt;br&gt;</b> @throws FatalException</u>
ICollateralTitleSearchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titesearch\ICollateralTitleSearchProcessAdapter	callTitleSearchClientProcessForCollateralUpdate		
ICollateralTitleSearchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titesearch\ICollateralTitleSearchProcessAdapter	callTitleSearchClientProcessForOnlineCancel		This method will be invoked when , the <b>&lt;b&gt;cancellation&lt;/b&gt;</b> request has been <b>&lt;b&gt;confirmed&lt;/b&gt;</b> from Valex. This will result in the request status being set to <u>cancelled&lt;/u&gt;. @fcb.param SessionContext,sessionContext, Holds the basic information about the user of the system. @fcb.param CollateralTitleSearchCancelConfirmationDTO, collateralTitleSearchCancelConfirmationDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance <b>&lt;br&gt;</b> of TransactionStatus contains the information whether the transaction is success\failure. <b>&lt;br&gt;</b> @throws FatalException</u>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralTitleSearchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchProcessAdapter	checkIfReportIsLastForARequest		
ICollateralTitleSearchProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchProcessAdapter	checkIfReportIsLastForARequest		
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		The following method is used to build internal account component preference DTO. The following method returns you a list of accounts that have been stored at the component preference item level. @param componentPreference @return @throws FatalException
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		This method is used to validate the inter product transfer.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		This method is called to validate the Hiso related validation.
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context Indicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when settlement engine should initiate any execution on its own so that the funds can be made available on time. For e.g., for a Loans PDC for arrears repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrears Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. The module adapter will have to determine the facts required by e-Pricing to determine mode usage charges. E.g., for payments, the network related charges can be calculated only if the network id as a fact is known @param context @param amount @param details @param direction @param transactionType @throws FatalException
ISettlementDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementDemandDepositAdapter	fetchAccountNoUsingCounterParty		This method is used to perform account related checks.
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	sendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	resendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	resendSMS		
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	validate		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOAAMWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOAAMWebServiceAdapter	validate		
ICommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ICommonServiceAdapter	fetchRateForCurrentBranch	This method will invoke fetchRateForCurrentBranch from CommonService which will return conversion rate for input currency pair. @param,MI,String,currency1, ,This parameter represents currency1 @param,MI,String,currency2, ,This parameter represents currency2 @return float - This method returns conversion rate.	This operation checks the both account and transaction rate change by the  user of the API against allowed exchange rate variance limit of the user. @fcb.param MI,String,taskCode, , task code. @fcb.param MI,String,localCurrencyCode, , Currency code. @fcb.param MI,String,postingDate, , The Date on which transaction is posted. @fcb.param MI,ITransactionLimit,transactionLimit, ,Defined transaction limits. @fcb.param MI,float,newRate, , Changed Currency rate by the user. @return int Indicates whether authorization required or not. @throws FatalException @modelguid {CED2B0DD-F523-4A72-8654-3BD7051868B2}
ICommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ICommonServiceAdapter	fetchRateForCurrentBranch	This method will invoke fetchRateForCurrentBranch from CommonService which will return conversion rate for input currency pair. @param,MI,String,currency1, ,This parameter represents currency1 @param,MI,String,currency2, ,This parameter represents currency2 @return float - This method returns conversion rate.	This method will invoke fetchRateForCurrentBranch from CommonService which will return conversion rate for input currency pair. @param,MI,String,currency1, ,This parameter represents currency1 @param,MI,String,currency2, ,This parameter represents currency2 @return float - This method returns conversion rate.
ICommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ICommonServiceAdapter	addApprovals	This method adds approval comments.  @param sessionContext @param approvalCommentDTOs @return TransactionStatus @throws	This method adds approval comments.  @param sessionContext @param approvalCommentDTOs @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
ICommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ICommonServiceAdapter	addApprovalComments	This method adds approval comments.  @param sessionContext @param approvalCommentDTOs @return TransactionStatus @throws FatalException	This method adds approval comments.  @param sessionContext @param approvalCommentDTOs @return TransactionStatus @throws FatalException
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isPartyValid	This method is used to validate whether given party id is valid or not @param partyId @return	This method validate party id. @param partyId
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isPartyValid	This method is used to validate whether given party id is valid or not @param partyId @return	The parties present in the linkages are also verified.
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isPartyValid	This method is used to validate whether given party id is valid or not @param partyId @return	The parties present in the linkages are also verified.
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isPartyValid	This method is used to validate whether given party id is valid or not @param partyId @return	Checks whether the party id provided is valid or not.
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isPartyValid	This method is used to validate whether given party id is valid or not @param partyId @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchRelatedParty	This method is used to fetch related party for given partyId list @param partyId @return	This method is used to obtain the array of related parties associate with party along with the other parties associated with corresponding borrowingEntities @param appContext @param partyList @throws FatalException
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchRelatedParty	This method is used to fetch related party for given partyId list @param partyId @return	This method fetches all the related parties for the given set of input party id's where the relationship allows to share exposure. @param partyId @return @throws FatalException
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyDeterminantValue	This method is to fetch determinant value of party @param partyId @return	This method is used to fetch the FactValues for party @param exposurefactslst @param modulecode @param party Id
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyWarningIndicators	This method fetches all the warning indicators that are the for the party @param partyIdArray @return @throws FatalException	This method is used to fetch the party details from the party module. Warning indicator is also fetched along with the party details. Party Legal name is also fetched and set here. @param partyIdArray @return FacilityPartyDetailsDTO[] @throws FatalException
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyWarningIndicators	This method fetches all the warning indicators that are the for the party @param partyIdArray @return @throws FatalException	This method will call the party module through an adapter and fetch all the warning indicators/memos present for the party @param partyIdArray @return @throws com.ofss.fc.infra.exception.FatalException
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyName	This method is to fetch partNname of the party @param partyId @return @throws FatalException	This method is used to create the charge records for collaterals which are only supporting and does not have any direct linkage to the facility. @param referenceId @param collateralId @param linkageDtlsList @param facilityPartyDetailsDTOArray @param chargeDetailsSet

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyName	This method is to fetch partNname of the party @param partyId @return @throws FatalException	This method is used to create the charge detail record required for the credit exposure documentation. @param linkageDtlsList
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyName	This method is to fetch partNname of the party @param partyId @return @throws FatalException	This represents the fetchBorrowingEntity method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BorrowingEntityDTO,borrowingEntityDTO, ,borrowingEntityDTO TODO: Explain the borrowingEntityDTO above @return {@link com.ofss.fc.app.elcm.dto.limits.borrowingapp.BorrowingEntityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyName	This method is to fetch partNname of the party @param partyId @return @throws FatalException	This represents the fetchByVersion method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BorrowingEntityDTO,borrowingEntityDTO, ,borrowingEntityDTO TODO: Explain the borrowingEntityDTO above @return {@link DomainObjectDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyName	This method is to fetch partNname of the party @param partyId @return @throws FatalException	
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchPartyWarningIndicatorsForExposure	This method is used to fetch the party details wrt to the warning indicator and risk score. Based on the isRiskScoreRequired parameter system will fetch the risk score of the party. @param partyIdArray @param isRiskScoreRequired @since 2.3 @return @throws FatalException	This method is used to fetch the party details from the party module. Warning indicator is also fetched along with the party details. Party Legal name is also fetched and set here. @param partyIdArray @return FacilityPartyDetailsDTO[] @throws FatalException
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isValidParty		This method is to return guarantee type of parties for the given facilities @param facilityIdArray @param isProposedFacilityRequired @param isBEPartyRequired @return @throws FatalException
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	isValidParty		This method is used to validate whether given party id is valid or not @param partyId @return
ILimitsPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchAllPartiesOfAccount	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationShip that exists @param accountId @return	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationShip that exists @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	fetchAllPartiesOfAccount	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationship that exists @param accountId @return	
ILimitsParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPartyAdapter	populateBorrowingEntityDTO		This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException
IOriginationMDMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This service will populate the Applicant Data DTO using the party identifier of the applicant.  The party details are fetched using the MDM service. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDTO applicantDTO, , The applicant information for which details are being populated. @return ApplicantDataDTO, Returns applicant information. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This service will populate the Applicant Data DTO from Applicant DTO @param sessionContext @param applicantDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This method is used to fetch a list of applicants by passing an applicationId. NOTE: Caller must make sure that the input applicationId is validated. The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception     {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052   This error occurs when:  The application identifier is invalid and no application details are found for the same. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This method is used to fetch a list of applicants and affiliates for the input applicationId.   NOTE: Caller must make sure that the input applicationId is validated.   The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. The related affiliates are also added to the list of applicants. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception     {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052   This error occurs when:  The application identifier is invalid and no application details are found for the same. 



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This service fetches party details and populates factsMap to execute condition/covenant policy @param sessionContext @param applicantDecisionDTO @param borrowingEntityId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	The method will construct facts for the primary applicant in the application @fcb.param MI, applicationId - Application details which will be used to construct the factsDTO. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	Construct facts for part from Origination fact application (MDM party facts) @param partyFacts @param applicantId @param partyId @param sessionContext @return String @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	This service will fetch the guarantors for a particular facility. It will check with LCM for the guarantee collaterals linked to this facility and then for all those owners fetch the party details from MDM @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return MDMPartiesInquiryResponse , Response containing details of all guarantors for the facility. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchMDMPartyDetails	This service will fetch the detailed party information @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, internalPartyId, , Indicates the party identifier from internal party module. @fcb.param MI, String, externalPartyId, , Indicates the party identifier from OCH (oracle customer hub). @fcb.param MI, PartyType, partyType, , Indicates the party type. @return MDMPartyDTO, indicates the detailed party information for the identifier passed @throws FatalException	
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchPartyDetailedInformation		This method creates an array of {@link ApplicantDataDTO} which contains information about the applicants which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link ApplicantDataDTO} @throws FatalException
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchPartyDetailedInformation		This method will fetch the party to party relationships between all parties for the given borrowing entity id @param sessionContext the {@link SessionContext} @param borrowingEntityId the borrowing entity id @return an array of {@link PartyToPartyRelationshipDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchPartyDetailedInformation		<p>This service will confirm structure solution for lending product group in a submission. &lt;br&gt; Various operations performed during this service are as follows: &lt;br&gt; &lt;ol&gt; &lt;li&gt;The facility structure is validated as per the guidelines. {@link LendingApplicationHelper#checkFacilityConformity}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Validations for confirming structure solution are validated. {@link LendingApplicationHelper#validateConfirmActionOnStructureSolution}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if ticket number for Origination fees is present. If yes, defer all the uncollected fees for that ticket number else create a new pricing plan, fetch and defer the origination fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if product group is setup under a bundle and ticket number for bundle fees is present. If yes, defer all the fees for that ticket number else create a new pricing plan, fetch and defer the bundle fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;In case of Amended facility, check if settlement is required or not (irrespective of the settlement flag on product group and update the same) {@link LendingApplicationHelper#checkIfSettlementRequiredForPG}&lt;/li&gt;&lt;br&gt; &lt;li&gt;For Loan product groups, update party id, borrowing entity and facility details for loan simulations&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check for any rates and fee negotiations. If present, send for discretionary pricing approval&lt;/li&gt;&lt;br&gt; &lt;/ol&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @return LendingAccountCreationResponse, this class returns</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the LendingAccountCreationResponse object,  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#KYC_DETAILS_NOT_FO  UND} : FC_OR_PI_0001 &lt;br&gt; This check is part of  Pre-create account validation. &lt;br&gt; This exception is  thrown if KYC has not been confirmed for all parties  &lt;br&gt; applying to the product group for which account  is being opened. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_DPA_APPROVAL} :  FC_OR_DE_0082 &lt;br&gt; This exception is raised as an  overridable exception when pricing rates and fees are  negotiated and require approval. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_AMOUNT_  GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception  occurs when one or more sub facilities are configured  and the requested amount for these sub-lines exceed  their parent facility's (can be main facility or another  sub facility) amount. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_TERM_GRE  ATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs  when one or more sub facilities are configured and the  requested term for these sub-lines exceed their parent  facility's (can be main facility or another sub facility)  term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_  SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#LVR_THRESHOLD_BREA  CHED} : FC_OR_SS_0006 &lt;br&gt; This exception occurs  when the loan to value ratio is breached beyond the  limit allowed for a product group. This threshold  value is maintained in PMU in credit policy template</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>for the product group &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under a single facility or sub-facility is greater than requested amount of that facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER_THAN_MAIN_FACILITY} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the facility structure exceed the total amount of main facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#ACCT_NOT_CONFIGURED} : FC_OR_SS_0015 &lt;br&gt; This exception occurs when structure solution is being confirmed with no account configured under the facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FETCH_PRICE_OFFERINGS_FAILED} : FC_OR_SS_0016 &lt;br&gt; This exception occurs when fetching of price recommendations for origination fees from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PPC_CREATION_FAILED} : FC_OR_SS_0017 &lt;br&gt; This exception occurs when creation of pricing plan for origination fees in pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#GET_PRICE_ON_EVENT_FAILED} : FC_OR_SS_0018 &lt;br&gt; This exception occurs when fetching price for events in origination (OR_NEW_FACILITY or OR_AMEND_FACILITY) from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OD_FIXED_TERM_GREATER} : FC_OR_SS_0019 &lt;br&gt; This exception occurs when fixed term entered for overdraft product groups</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>exceed the parent requested limit term.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#APPLICANTS HOLDER_PREFERENCES_INCORRECT} : FC_OR_SS_0021&lt;br&gt; This exception occurs when user configures the account configuration in structure solution and then goes back to the product details page and changes the applicants applying for the product group.&lt;br&gt; The user is prompted to check the account configuration again to change the account holder preference for changed applicants.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_FEE_NEGOTIATION} : FC_PR_FEE_OVR &lt;br&gt; This exception occurs any of the rates or fees are negotiated.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#COLLATERAL_VALUATION_INCOMPLETE} : FC_OR_CL_0001 &lt;br&gt; This exception occurs if NCCP is yes and valuation for any collaterals linked to the facility is not yet complete.&lt;br&gt;</p>
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchPartyDetailedInformation		
IOriginationMDMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAadapter	fetchPartyDetailedInformation		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationMDMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchConductDetails	@fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String[], partyIds, , the array of the party ids. @return PartyConductDetailsResponse, this object will contain all the PartyConductDetails @throws FatalException	<<2.3>>  This method is used to retrieve and save the PartyConductDetailsDTOs for one given ApplicationDecisionDTO @param applicationDecisionDTO @throws FatalException
IOriginationMDMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationMDMAAdapter	fetchConductDetails	@fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String[], partyIds, , the array of the party ids. @return PartyConductDetailsResponse, this object will contain all the PartyConductDetails @throws FatalException	@fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String[], partyIds, , the array of the party ids. @return PartyConductDetailsResponse, this object will contain all the PartyConductDetails @throws FatalException
IApplicationEnterpriseRoleLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IApplicationEnterpriseRoleLoaderAdapter	loadMappedRoles		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IApplicationEnterpriseRoleLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IApplicationEnterpriseRoleLoaderAdapter	loadMappedRoles		Initialize all security related singletons @param sessionContext @throws FatalException
ILoanAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountAdapter	addOffsetAccount	This method is used to add a new Loan Account Principal Offset.	This method is used by loan module to add the offset account provided by account module. @param offsetAccountDTO @throws FatalException
ILoanAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountAdapter	deleteOffsetAccount	This method is used to delete an existing Loan Account Principal Offset.</br>	This method is used by loan module to delete an existing the offset account linkage. @param offsetAccountDTO
ILoanAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountAdapter	linkLoanAccountToOtherBranch	This method is used to link LoanAccountBranchTransferApplication for Loan Account Branch Transfer @param sessionContext @param loanAccountBranchTransferDetails DTO @throws FatalException	This method is being used to call loan account transfer from Common Account Transfer. The operation linkLoanAccountToOtherBranch is being called from AccountBranchTransferApplicationService API if module type for the account is LOAN. @param sessionContext @param addAccountBranchTransferDetailsDTO @throws FatalException
ILoanAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountAdapter	logInterestBenefitAction	This method is used to log interest benefit action. @param accountId	This method is used to log interest benefit action. @param accountId
ILoanAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountAdapter	logActivityForBatch	' Logging the alert @param applicationContext @param offsetAccountDTO @param ActivityId	This method is used to raise alert for offset account @param applicationContext @param offsetAccountDTO @param loanActivityId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDDDAAdapter	reduceTdSweepinLienAmount	This method calls DemandDepositAccountApplicationService to reduce td sweepin lien amount marked on Beneficiary account. @param accountId Beneficiary deposit account id @param lienAmountDTO Lien amount to be reduced @param datesDTO {@link TransactionDatesDTO} DTO object containing posting, value and transaction dates @return {@link SweepinLienRemovalResponse} containing details of Sweep in Lien removal	This method calls DemandDepositAccountApplicationService to reduce td sweepin lien amount marked on Beneficiary account. @param accountId Beneficiary account id @param lienAmountDTO Lien amount to be reduced @param datesDTO {@link TransactionDatesDTO} DTO object containing posting, value and transaction dates @return {@link SweepinLienRemovalResponse} containing details of Sweep in Lien removal @see com.ofss.fc.app.adapter.td.ITDDDAAdapter#reduceTdSweepinLienAmount(java.lang.String, com.ofss.fc.datatype.dto.MoneyDTO, com.ofss.fc.app.account.dto.transaction.TransactionDatesDTO)
ITDDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDDDAAdapter	fetchBasicDetails	The method will fetch the basic details of given casa account. @param accountId casa account id @return {@link BasicInquiryDetailsDTO} Object containing details of an account	This method populates component preference dto for settlement @param mode {@link SettlementModeType} Enum for settlement mode @param accountNo account id of term deposit @param currencyCode Currency code @param redemptionAmount the amount which is to be redeemed @return {@link ComponentPreferenceDTO}
ITDDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDDDAAdapter	fetchBasicDetails	The method will fetch the basic details of given casa account. @param accountId casa account id @return {@link BasicInquiryDetailsDTO} Object containing details of an account	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDDDAAdapter	fetchBasicDetails	The method will fetch the basic details of given casa account. @param accountId casa account id @return {@link BasicInquiryDetailsDTO} Object containing details of an account	This method is used to Payin a Term Deposit from a Demand Deposit Account using minimum input parameters. The other mandatory input will be derived from the respective product definitions. @param accountId The account Id of the Term Deposit account @param providerAccountId The account Id of the Demand Deposit account that will fund the Payin amount @param payinAmount The actual Payin amount @param term The term for which the deposit is to be created. This is optional, if null the minimum term from the product definition will be taken. @return @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchOperatingInstructions	This service will fetch all the operating instructions financial as well as non financial instructions for the account passed. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId , , This represents the Account Id of the Deposit Account already opened. @fcb.param MI,OperatingInstructionCategoryType, instructionType, , This represents the instruction category type FINANCIAL or NON FINANCIAL @throws FatalException	This service will fetch account configurations done for a CASA account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g.: Internet banking, Debit Card, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the account configuration details for demand deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchOperatingInstructions	This service will fetch all the operating instructions financial as well as non financial instructions for the account passed. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId , , This represents the Account Id of the Deposit Account already opened. @fcb.param MI,OperatingInstructionCategoryType, instructionType, , This represents the instruction category type FINANCIAL or NON FINANCIAL @throws FatalException	This service will fetch account configurations done for a TD account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g.: Internet banking, Phone banking, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the account configuration details for term deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAccountAddressLinkages	This service will fetch the existing account address linkages @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.   @fcb.param MI,String, accountId , , This represents the Account Id of the Deposit Account already opened.   @fcb.param MI, AccountType, accountType, , Indicates the type of account   @throws FatalException	This service will fetch account configurations done for a CASA account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g.: Internet banking, Debit Card, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.   @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the account configuration details for demand deposit products.   @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAccountAddressLinkages	This service will fetch the existing account address linkages @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.   @fcb.param MI,String, accountId , , This represents the Account Id of the Deposit Account already opened.  @fcb.param MI, AccountType, accountType, , Indicates the type of account   @throws FatalException	This service will fetch account configurations done for a TD account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g.: Internet banking, Phone banking, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the account configuration details for term deposit products.   @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	checkAndLevyFees	This method will persist the settlement details for the collected fees and raise accounting Event. @param priceFeatureDTO @param applicationRequestType @param componentPreferenceDTO @throws FatalException	This method is called on save of collect fees. The following activities will be carried out in sequence:   1. Call the pricing service to persist the deferred and waived fees.  2. Call the settlement module to collect fees and raise accounting event for the fees to be collected. Component Preference DTO can be null in case there is no fees to be collected.   @param sessionContext @param priceFeatureDTO @param componentPreferenceDTO @param lendingProductGroupLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	checkAnd LevyFees	This method will persist the settlement details for the collected fees and raise accounting Event. @param priceFeatureDTO @param applicationRequestType @param componentPreferenceDTO @throws FatalException	This method is called on withdraw submission. The following activities will be carried out in sequence: 1. Call the pricing service to persist the deferred fees if any. 2. Call the settlement module to collect fees and raise accounting event for the fees to be collected. Component Preference DTO can be null in case there is no fees to be collected. @param sessionContext @param priceFeatureDTO @param componentPreferenceDTO @param submissionId @return @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	generateBundleReferenceId	This service will call the account module to generate the bundle reference Id. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @return String, Indicates the generated bundle reference Id @throws FatalException	This service will be called on bundle recommendation selection or change during post assessment structure solution. It will perform the following-  i) Validate if existing bundle can be cancelled and new bundle can be linked to the submission  ii) Cancel any existing bundle linked with the submission  iii) Subscribe bundle for existing accounts   iv) Save bundle details for the submission   v) Stamp bundle id for offers linked to the bundle   vi) Update PG as bundled PG if any offer under the PG is linked to bundle   If bundle reference id of new bundle is same as existing bundle then only existing data will be updated. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.app.origination.dto.core.bundle.BundleDetailsResponse} This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed and an instance of the BundleDetailsDTO which contains all information of the created bundle. @throws FatalException @exception&lt;br&gt; {@link OriginationErrorConstants#BUNDLE_SELECTION_NOT_ALLOWED} : FC_OR_BD_0011&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the bundled PG.&lt;br&gt; @exception&lt;br&gt; {@link OriginationErrorConstants#BUNDLE_CANCELLATION_NOT_ALLOWED} : FC_OR_BD_0010&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the PG linked to existing bundle.&lt;br&gt;</p>
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	generateBundleReferenceId	This service will call the account module to generate the bundle reference Id. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @return String, Indicates the generated bundle reference Id @throws FatalException	This service will call the account module to generate the bundle reference Id. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @return String, Indicates the generated bundle reference Id @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	generateBundleReferenceId	This service will call the account module to generate the bundle reference Id. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @return String, Indicates the generated bundle reference Id @throws FatalException	
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	inquireAccount	This service will fetch the account details @throws FatalException	Verify the account limits  1. Each account limit should be less than parent limit.  2. Each account term should be less than parent term.  3. Sum of all accounts under a facility should be less than or equal to parent limit.   4. Sum of all accounts configured under a facility of type non revolving should be less than or   equal to main facility limit and also conform to its own parent limit (same as 3 above).   5. OD Fixed Term should be less than or equal to its parent term.     Also, values for DPA (UDM, fees negotiations are read from each account configuration and set with the highest values. @param sessionContext @param creditAccountDTOs @param facilityId @param parentLimit @param parentTerm @param solutionDTO @return @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	inquireAccount	This service will fetch the account details @throws FatalException	This method populates the approved amount, available amount and account title for each loan offer inside each PG in the application. @param sessionContext @param productGroupLinkage @param settlementRef @param offerReferences @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	inquireAccount	This service will fetch the account details @throws FatalException	Calculate loan available amount for amend flow Available amount = Current Sanctioned amount - Total disbursed amount of existing account @param sessionContext @param approvedAmount @param existingAccountNumber @param currency @return MoneyDTO @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	inquireAccount	This service will fetch the account details @throws FatalException	This service inquires details of the existing account based on given account ids. Given account ids can be long to different account type. We'll call account service to inquire account based on account id, internally, they will inquire the P2A to get the account type and inquire necessary module on the account details. @fcb.param accountIds @return ExistingAccountInquiryResponse
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	dedupeForBundleOwners	This method validates whether the given list of parties has already subscribed for the same product bundle which they have subscribed in the current submission. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @param applicantMap @param bundleReferenceId @param bundleId @param offerId @param pricingTicketNumber @throws FatalException	This service is called from validateSubmission. If the submission has opted product bundle then it validates whether all the applicants under this submission is existing customer. If all are existing customers, then itChecks whether the same set of applicants already holding the same product bundle. If the validation fails it should add an exception to validation error with proper error message @param sessionContext @param submissionInputDTO @param bundleId @param bundleReferenceId @param nominatedOfferId @param pricingTicketNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	logBundleExceptionReport	This service will log bundle exception report details. In case of bundle subscription, if a submission, application or a product group is withdrawn and not all mandatory accounts are opened, exception is logged. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, bundleId, , Indicates the bundle identifier. @fcb.param MI, String, bundleReferenceId, , Indicates the bundle reference identifier @fcb.param MI, String, nominatedOfferId, , Indicates the offer identifier nominated for bundle fee. @fcb.param MI, String, applicationBranch, , Indicates the application branch @fcb.param MI, String, withDrawComments, , Indicates the comments for withdrawing submission, application or product group @fcb.param MI,String, productClass, , Indicates the type of product class to which the product group belongs @throws FatalException	This service will mark the input product group as withdrawn or rejected. If the whole submission has been withdrawn, all product groups within the submission will be withdrawn.  If a single product group is rejected (by rejecting any document at product group level) or application is rejected (by rejecting the offer letter), the product group state is marked as REJECTED.   Facility details are deleted in both the above cases  If the product group is a part of the bundle, a bundle exception log is generated and saved.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO productGroupLinkageDTO, , Indicates the product group being withdrawn or rejected. @fcb.param MI,String event,Indicates the corresponding event set by the caller @fcb.param CI,String nominatedOfferId, , Indicates the offer identifier for the product group nominated for bundle fees. This is applicable only if product group is part of a bundle @fcb.param CI,String applicationBranch, ,Indicates the application branch. This is applicable only if product group is part of a bundle @fcb.param OI,String withDrawComments, , Indicates the reason for withdrawal of product group. This is applicable only if product group is part of a bundle @fcb.param MI,boolean isMarkedForExpiry,Indicates the submission is marked for expiry @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedPartyDetails	This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	addAssociatedParty	This service will persist the associated party details @param SessionContext @param AssociatedPartyDTO [] @return AssociatedPartyDetailsResponse @throws FatalException	This service will persist the associated party details @param SessionContext @param AssociatedPartyDTO [] @return AssociatedPartyDetailsResponse @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	addAssociatedParty	This service will persist the associated party details @param SessionContext @param AssociatedPartyDTO [] @return AssociatedPartyDetailsResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	addAssociatedParty	This service will persist the associated party details @param SessionContext @param AssociatedPartyDTO [] @return AssociatedPartyDetailsResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	This service will update the Bundle Reference identifier for the list of applicants  applying for the bundle. The Bundle Reference identifier is generated upfront and is  used as the arrangement number for pricing plan created for product group nominated   for collection of bundle subscription fees. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Map<String,ApplicantDTO>,applicantMap, , Indicates the map of applicants with their party Ids as the key. @fcb.param MI, String, bundleReferenceId, , Indicates the bundle reference identifier generated. @fcb.param MI, String, bundleId, , Indicates the bundle identifier @fcb.param MI, String, offerId, , Indicates the nominated offer identifier @fcb.param MI, String, pricingTicketNumber, , Indicates the pricing ticket number under which the subscription fee details is to be fetched and charged. @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	This service will unsubscribe the bundle for the parties @param applicationContext @param bundleManagementDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	This service will subscribe/unsubscribe accounts or new offers to/from a bundle. If pricing ticket number is not set for bundle fees then this service will first create the pricing plan. @param sessionContext @param bundleDetailsDTO @param productDetails @param valueDate
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	subscribeBundle	This service will create new bundle if not present and link the accounts and parties passed as part of bundleSubscriptionDTO with the bundle @param sessionContext @param bundleSubscriptionDTO	This service will cancel already subscribed bundle linked to the submission.   It will be called during following scenarios - i) Bundle is subscribed during submission creation but cancelled/changed during edit submission  ii)If user subscribes for a (fully qualified) bundle through post credit assessment screen, and then if he cancels/changes the bundle This service will perform the following actions-   i) If the bundle was new then call bundle unsubscribe service to unsubscribe the bundle for the parties  ii) If the bundle was already existing then call bundle delinkAccount service to delink the accounts linked during this submission to the bundle  @param sessionContext @param submissionId @param bundleReferenceId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	delinkAccount	This service will unlink the accounts passed from the bundle @param applicationContext @param bundleReferenceId @param accountIdArray @throws FatalException	This service will cancel already subscribed bundle linked to the submission.   It will be called during following scenarios - i) Bundle is subscribed during submission creation but cancelled/changed during edit submission  ii)If user subscribes for a (fully qualified) bundle through post credit assessment screen, and then if he cancels/changes the bundle This service will perform the following actions-   i) If the bundle was new then call bundle unsubscribe service to unsubscribe the bundle for the parties  ii) If the bundle was already existing then call bundle delinkAccount service to delink the accounts linked during this submission to the bundle  @param sessionContext @param submissionId @param bundleReferenceId @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundleOwners		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundleOwners		This service will validate if the owners for chosen offers are eligible to be linked to the bundle

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		<p>This service will validate and save the bundle details in Origination. &lt;br&gt; The following validations are performed - &lt;br&gt; i) If a new PG needs to be created because of bundle. This scenario might arise if offer selected in bundle is for a PG which is not already present in the submission or if the PG is fulfilled or withdrawn.&lt;br&gt; ii) Bundle subscription fee treatment is Capitalize for the offer nominated for bundle fees but the Capitalize fees has not been selected for the product group during submission creation.&lt;br&gt; iii) All PG's with bundled offers should be Structure solution stage.&lt;br&gt; iv) In case of change of bundle, all PG's with offers linked to existing bundle should be Structure solution stage.&lt;br&gt; If the above checks are passed selected bundle will be persisted in Origination.&lt;br&gt; If submission has already linked bundle , the existing bundle details will be moved to history table. It will also create the product groups and create/update the offer references modified because of bundle selection.&lt;br&gt;This service will get called on Confirm of bundle recommendation panel in post assessment structure solution. @param sessionContext @param submissionInputDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		<p>This service will be called when user subscribes the bundle during post assessment structure solution. It will perform the following-</p> <ul style="list-style-type: none"> <li>i) Validate if the bundle can be subscribed</li> <li>ii) Subscribe bundle for existing accounts and new offers</li> <li>iii) Save or update bundle details for the submission</li> <li>iv) Log exception in bundle module for previously subscribed bundles for the submission and update their status as UNSUBSCRIBED in origination</li> <li>v) Stamp bundle id for offers linked to the bundle</li> <li>vi) Update PG as bundled PG if any offer under the PG is linked to bundle</li> </ul> <p>If bundle reference id of new bundle is same as existing bundle then only existing data will be updated.</p> <p><b>@fcb.param</b>  MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><b>@fcb.param</b> MI,SubmissionInputDTO, submissionInputDTO, This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader <a href="#">{@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO}</a> an array of ApplicantData <a href="#">{@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO}</a> and an array of ProductDetails <a href="#">{@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO}</a>. Refer to <a href="#">{@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO}</a> for more details. <b>@return</b> <a href="#">{@link com.ofss.fc.app.origination.dto.core.bundle.BundleDetailsResponse}</a> This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>indicating if the invocation succeeded or failed and an instance of the BundleDetailsDTO which contains all information of the created bundle. @throws FatalException @exception&lt;br&gt;{@link OriginationErrorConstants#BUNDLE_SELECTION_NOT_ALLOWED} : FC_OR_BD_0011&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the bundled PG.&lt;br&gt; @exception&lt;br&gt;{@link OriginationErrorConstants#BUNDLE_CANCELLATION_NOT_ALLOWED} : FC_OR_BD_0010&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the PG linked to existing bundle.&lt;br&gt;</p>
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		<p>This service will check and return whether edit submission is required due to bundle selection. It will be called during post assessment structure solution task if any recommended bundle is chosen by the user.&lt;br&gt; The following conditions will be checked-&lt;br&gt; i) If a new PG needs to be created because of bundle. This scenario might arise if offer selected in bundle is for a PG which is not already present in the submission or if the PG is fulfilled or withdrawn.&lt;br&gt; ii) Bundle subscription fee treatment is Capitalize for the offer nominated for bundle fees but the Capitalize fees has not been selected for the product group during submission creation.&lt;br&gt; iii) All PG's with bundled offers should be Structure solution stage.&lt;br&gt; iv) In case of change of bundle, all PG's with offers linked to existing bundle should be Structure solution stage.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.app.origination.dto.core.bundle.BundleEditResponse} This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed and a flag to indicate if Edit submission is required. @throws FatalException @exception&lt;br&gt; {@link OriginationErrorConstants#BUNDLE_REQUIRES_NEW_PG} : FC_OR_BD_0013&lt;br&gt; This exception is thrown if new PG is required to be created due to bundle selection.&lt;br&gt; @exception&lt;br&gt; {@link OriginationErrorConstants#CAPITALIZE_BUNDLE_FEES} : FC_OR_BD_0009&lt;br&gt; This exception is thrown if bundle subscription fee treatment is Capitalize but Fee Capitalization has not been selected for the PG.&lt;br&gt; @exception&lt;br&gt; {@link OriginationErrorConstants#BUNDLE_SELECTION_NOT_ALLOWED} : FC_OR_BD_0011&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the bundled PG.&lt;br&gt; @exception&lt;br&gt; {@link OriginationErrorConstants#BUNDLE_CANCELLATI</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ON_NOT_ALLOWED) : FC_OR_BD_0010  This exception is thrown if structure solution is confirmed for any of the PG linked to existing bundle. 
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		This service will validate if the owners for chosen offers are eligible to be linked to the bundle



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		Perform bundle module validations
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		This method is called from validateSubmission. If the submission is of product bundle type then it validates whether all the mandatory product groups for the product bundle is applied or not. If the validation fails it should add an exception to validation error with proper error message @param submissionInputDTO @param bundleId @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateBundle		This method is called from the service validateSubmission. It checks for the following conditions.  1. If the product bundle is selected, then it validates whether all the product groups and the offers are opted for bundle  2. one of the product group should be opted for bundle fees  If any validation fails it should add an exception to validation error with proper error message @param submissionInputDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	logBundleException		<p>This service will be called when user subscribes the bundle during post assessment structure solution. It will perform the following-</p> <ul style="list-style-type: none"> <li>i) Validate if the bundle can be subscribed</li> <li>ii) Subscribe bundle for existing accounts and new offers</li> <li>iii) Save or update bundle details for the submission</li> <li>iv) Log exception in bundle module for previously subscribed bundles for the submission and update their status as UNSUBSCRIBED in origination</li> <li>v) Stamp bundle id for offers linked to the bundle</li> <li>vi) Update PG as bundled PG if any offer under the PG is linked to bundle</li> </ul> <p>If bundle reference id of new bundle is same as existing bundle then only existing data will be updated.</p> <p><b>@fcb.param</b>  MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><b>@fcb.param</b> MI,SubmissionInputDTO, submissionInputDTO, This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {<a href="#">@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO</a>} an array of ApplicantData {<a href="#">@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO</a>} and an array of ProductDetails {<a href="#">@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO</a>}. Refer to {<a href="#">@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO</a>} for more details. <b>@return</b> {<a href="#">@link com.ofss.fc.app.origination.dto.core.bundle.BundleDetailsResponse</a>} This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>indicating if the invocation succeeded or failed and an instance of the BundleDetailsDTO which contains all information of the created bundle. @throws FatalException @exception&lt;br&gt;{@link OriginationErrorConstants#BUNDLE_SELECTION_NOT_ALLOWED} : FC_OR_BD_0011&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the bundled PG.&lt;br&gt; @exception&lt;br&gt;{@link OriginationErrorConstants#BUNDLE_CANCELLATION_NOT_ALLOWED} : FC_OR_BD_0010&lt;br&gt; This exception is thrown if structure solution is confirmed for any of the PG linked to existing bundle.&lt;br&gt;</p>
IOrginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOrginationAccountAdapter	logBundleException		<p>This service will cancel existing bundle already linked to a submission. It will also be called during update of bundle detail. If new bundle reference id is same as existing bundle reference id then bundle will not be cancelled. &lt;br&gt;This service will perform the following actions- &lt;br&gt; i) Check if any bundle is linked with submission &lt;br&gt; ii) If yes, check if different bundle has been selected or user has cancelled bundle&lt;br&gt; iii)If existing bundle is present, move the bundle details to bundle history &lt;br&gt; iv) Delete bundle details from main tables &lt;br&gt; v) Intimate bundle module by logging failure exception for unfulfilled offers linked to bundle @param sessionContext @param submissionId @param bundleReferenceId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	logBundleException		<p>This service will mark the input product group as withdrawn or rejected. If the whole submission has been withdrawn, all product groups within the submission will be withdrawn.&lt;br&gt; If a single product group is rejected (by rejecting any document at product group level) or application is rejected (by rejecting the offer letter), the product group state is marked as REJECTED. &lt;br&gt; Facility details are deleted in both the above cases&lt;br&gt; If the product group is a part of the bundle, a bundle exception log is generated and saved. &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO productGroupLinkageDTO, , Indicates the product group being withdrawn or rejected. @fcb.param MI,String event,Indicates the corresponding event set by the caller @fcb.param CI,String nominatedOfferId, , Indicates the offer identifier for the product group nominated for bundle fees. This is applicable only if product group is part of a bundle @fcb.param CI,String applicationBranch, ,Indicates the application branch. This is applicable only if product group is part of a bundle @fcb.param OI,String withdrawComments, , Indicates the reason for withdrawal of product group. This is applicable only if product group is part of a bundle @fcb.param MI,boolean isMarkedForExpiry,Indicates the submission is marked for expiry @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateMandatoryBundleOffers		This service will validate if the mandatory offers and number of mandatory offers in the subset have been selected for the bundle @param sessionContext @param bundleOfferDTO @return
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateMandatoryBundleOffers		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchBundlesForParties		This service will be called on change of borrowers scenario to check if any bundles are linked to the accounts which are getting affected. If new parties are getting linked to accounts which are already part of bundle then the parties should be added as bundle owners as well. This service will return the bundle details and an array of bundled accounts which are impacted per bundle.  The following steps are performed-  i) All bundles with bundle owners as parties in the old parties list are fetched  ii) If bundle status is Active or Inactive, check that at least one party is present in new parties list which was not present as existing Bundle owner  iii) Create list of credit accounts under the facility being amended which are part of the above bundles  iv) It will also create a list of bundle owners based on the union of new owners of accounts which are linked to bundle as well as existing bundle owners.   @param sessionContext @param facilityBundleDetailsRequest @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		This method fetches the bundle Recommendations. It performs the following <ol style="list-style-type: none"> <li>fetch subscribed bundles</li> <li>populate BundleRecommendationRequestDTO</li> <li>fetch recommended bundles</li> </ol> @param productDetailsDTOArray @param headerDTO @param applicantDataDTOArray @param sessionContext @return BundleRecommendationResponse
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		This method fetches the BundleRecommendation from the Account module. @param bundleRecommendationRequestDTO @param sessionContext @return BundleRecommendationResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		<p>This API performs change of Owners and change nomination of account during amend flow in bundles. The Amend of Bundle occurs based on action passed in the {@link AmendBundleDTO} . The two values that can be passed are</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;{@link AmendBundleAction#CHANGE_OWNERS}&lt;/li&gt;</li> <li>&lt;li&gt;{@link AmendBundleAction#DELINK_ACCOUNT}&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; &lt;br&gt; It performs following functions</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;For the reference id passed the corresponding bundle is fetched&lt;/li&gt;</li> <li>&lt;li&gt;If not found a new one is created&lt;/li&gt;</li> <li>&lt;li&gt;a list of party Id is created from the current submission which is being amended&lt;/li&gt;</li> <li>&lt;li&gt;If bundle action is set to amend the partyIds are set and ComponentPreferenceDTO is set to null&lt;/li&gt;</li> <li>&lt;li&gt;else the account which needs to be delinked is fetched and the account details are set with the updated componentPreferenceDTO which is populated from the settlement Taskflow&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;</p> <p>&lt;br&gt; Following are the Prerequisites for using this method...</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;{@link SubmissionCreationApplicationService#generateSubmissionId(SessionContext, com.ofss.fc.app.origination.dto.core.submission.HeaderDTO)} &lt;/li&gt;</li> <li>&lt;li&gt;{@link SubmissionCreationApplicationService#amendLendingFacility(SessionContext, String, String)} &lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @param sessionContext MI,SessionContext,sessionContext, This parameter</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>captures the basic information about the user of the system. The information is used by the system for authentication purposes. Session Context must contain the session context information such as user id, branch, branch code, channel etc. @param submissionId MI,String,submissionId, contains the submission id against which the bundles are managed. @param amendBundleDTO contains the following attributes</p> <ul style="list-style-type: none"> <li>&lt;li&gt;bundleReferenceId&lt;/li&gt;</li> <li>&lt;li&gt;amendBundleAction&lt;/li&gt;</li> <li>&lt;li&gt;accountDetails&lt;/li&gt;</li> <li>&lt;li&gt;nominatedAccountId&lt;/li&gt;</li> <li>&lt;li&gt;partyIds&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @return Transaction Status. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @throws {@link com.ofss.fc.infra.exception.FatalException} Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		<p>This function fetches the AccountDetailsDTO of the delinked Account @param amendBundleDTO @param nominatedAccountId @return</p>
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		Process ApplicationBatch data and create Application, ProductGroupLinkage and save Simulation related data @param sessionContext @param submissionInputDTO @param response @param applicantDataList @param applicationBatch @return @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	amendBundle		This service will update the party Id references using the Applicant Data DTO. In case of new parties onboarded, the applicant data dto is updated with the new party identifiers.  All other DTOs requiring party Ids are initially updated using applicant serial number when party Ids are not available. After onboarding process, the applicant serial numbers are replaced with the party Id from Applicant Data DTO. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc.It consists of header information such as application branch, channel, creation date, etc. , details of applicants applying for different product groups, collateral information in case of secured lending product groups. Relationships within the applicants for a submission are also present if provided. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		This method will refund the origination fees during edit/withdraw submission or during Application form prev and next click. This is required each time the pricing fees are refreshed and a new ticket number is generated. The fees of the previous ticket are not to be used or used for any inquiries further . @param applicationContext @param pricingTicketNo @param arrangementNo @param currency @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		This method populates the Funding table details for the Lending product groups present in the given {@link SubmissionInputDTO}. It iterates over the product details array and populates the funding table for lending product groups, if the funding table is applicable @param sessionContext @param submissionInputDTO @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		This method will refund the collected fees against the ticket number into suspense GL @param sessionContext @param pricingTicketNo @param arrangementNo @param currency @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		<p>This service will inquire the submission with the given application Id and update the status of submission to SUBMISSION_IN_PROGRESS. This service will be called from the process to revoke the withdraw operation for submission @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,submissionId, ,The id of the submission to be inquired @return An instance of {@link TransactionStatus} containing information about the submission. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_ALREADY_WITHDRAWN} : FC_OR_WA_0002 &lt;br&gt; This error occurs when:&lt;br&gt; A submission is already withdrawn and the user tries to cancel withdraw for such submission. This will happen when the same submission is being accessed through human task and application tracker.&lt;br&gt; The user inquires the submission through application tracker and then withdraws the submission from the human task and later returns to the tracker to cancel withdraw.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		<p>This service will be called during edit submission and check the following:</p> <ul style="list-style-type: none"> <li>1&gt; Fetch all lending product groups under a lending application.</li> <li>2&gt; Check if account opening has been done for all accounts within the product group. If done, mark the application for status update as PROCESSED.</li> <li>3&gt; If product group has not been rejected or withdrawn, check if product group linkage is present after the user has modified the submission. This is checked by comparing the old facility list and incoming new facility list.</li> <li>4&gt; If a facility that previously existed is not found in the incoming list, then delete the facility hierarchy from LCM. If LMI policy and/or auto refinance policies have been initiated for such a facility, cancel the policies.</li> <li>5&gt; Finally for applications containing processed as well as not processed product groups, delete the product group linkages that are not processed from the applications as these will be referred in the new applications generated in edit.</li> </ul> <p><b>@fcb.param</b> MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><b>@fcb.param</b> MILEndingApplication lendingApplication, , Indicates the lending application being checked for marking as PROCESSED.</p> <p><b>@fcb.param</b> MI,List&lt;String&gt; newFacilityList, , Indicates the incoming new facility list. <b>@return</b> Boolean, Whether the application is to be marked as PROCESSED. <b>@throws</b> FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		<ul style="list-style-type: none"> <li>This method populates the following details for a given lending product group linkage.</li> <li>1. Populates the necessary facts requires for the Pricing call.</li> <li>2. Generates the pricing ticket number by initiating the start dialogue method.</li> <li>3. Fetches the list of pricing fund items applicable for the given funding template id's.</li> <li>4. Make the pricing call with all necessary facts and fetches the pricing fees for the product group.</li> <li>5. Updates the pricing ticket number in the product group linkage.</li> <li>6. Checks whether the LMI is applicable for the product group. If applicable it populates the LMI funding details.</li> <li>7. Populate CCI amount</li> </ul> <p>LMI should be fetched after CapitalizationDetailsDTO being populated so that it will take into acc the capitalization amount</p> <p>@param sessionContext  @param submissionInputDTO  @param lendingPgLinkageDTO  @param creditFacilityDTOs  @return {@link FundingTableDetailDTO}  @throws FatalException</p>
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	refundCollectedFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	updateAccountVariationRequestStatus	This method is used to update the status {@link VariationRequestStatusType} for the Account Variation Request identified by the unique Variation Request Id. @param sessionContext the {@link SessionContext} @param variationRequestId the unique Variation Request Id @param status the new {@link VariationRequestStatusType} to be set @return the {@link TransactionStatus} @throws FatalException	This method is used to update the status {@link VariationRequestStatusType} for the Account Variation Request identified by the unique Variation Request Id. @param sessionContext the {@link SessionContext} @param variationRequestId the unique Variation Request Id @param status the new {@link VariationRequestStatusType} to be set @return the {@link TransactionStatus} @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	updateAccountVariationRequestStatus	This method is used to update the status {@link VariationRequestStatusType} for the Account Variation Request identified by the unique Variation Request Id. @param sessionContext the {@link SessionContext} @param variationRequestId the unique Variation Request Id @param status the new {@link VariationRequestStatusType} to be set @return the {@link TransactionStatus} @throws FatalException	
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	validateManageBundleOwners		This service will validate if delink of account is allowed for the bundle @param applicationContext @param bundleReferenceId @param accountDetailsDTOArray @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	unsubscribe Bundle	This service will unsubscribe the bundle for the parties @param applicationContext @param bundleManagementDTO @throws FatalException	This service will unsubscribe the bundle for the parties @param applicationContext @param bundleManagementDTO @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	unsubscribe Bundle	This service will unsubscribe the bundle for the parties @param applicationContext @param bundleManagementDTO @throws FatalException	This service will cancel already subscribed bundle linked to the submission.   It will be called during following scenarios - i) Bundle is subscribed during submission creation but cancelled/changed during edit submission  ii)If user subscribes for a (fully qualified) bundle through post credit assessment screen, and then if he cancels/changes the bundle This service will perform the following actions-   i) If the bundle was new then call bundle unsubscribe service to unsubscribe the bundle for the parties  ii) If the bundle was already existing then call bundle delinkAccount service to delink the accounts linked during this submission to the bundle  @param sessionContext @param submissionId @param bundleReferenceId @throws FatalException
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	updateAssociatedParty		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	updateAssociatedParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		This service will fetch the associated party details @param SessionContext @param AssociatedPartyDTO @return AssociatedPartyDetailsResponse @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchAssociatedParty		This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	deleteAssociatedParty		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	deleteAssociatedParty		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		
IOriginationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommendBundles		This method fetches the bundle Recommendations. It performs the following <ol style="list-style-type: none"> <li>fetch subscribed bundles</li> <li>populate BundleRecommendationRequestDTO</li> <li>fetch recommended bundles</li> </ol> @param productDetailsDTOArray @param headerDTO @param applicantDataDTOArray @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				sessionContext @return BundleRecommendationResponse
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	recommend Bundles		This method fetches the BundleRecommendation from the Account module. @param bundleRecommendationRequestDTO @param sessionContext @return BundleRecommendationResponse
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchUnAssociatedAccounts		
IOrigination AccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAccountAdapter	fetchUnAssociatedAccounts		This method populates the Bundle Account Reference DTO based on existing Accounts @param bundleDetailsDTO @param productGroupLinkageDTOArray @param applicantDataDTOArray @param submissionValueDate @param sessionContext @param submissionInputDTO
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	fetchLoanAccountsInformation	Service to return all the loans Account information to be displayed in SPIV  @param applicationContext @param partyId @return LoanAccountAttributesDTO	
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	fetchLoanAccountsInformation	Service to return all the loans Account information to be displayed in SPIV  @param applicationContext @param partyId @return LoanAccountAttributesDTO	Service to fetch all the Loans Account Information of the Party  @param sessionContext @param partyId @return LoanAccountInformationResponse @throws FatalException p

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	fetchLoanAccountsInformation	Service to return all the loans Account information to be displayed in SPIV  @param applicationContext @param partyId @return LoanAccountAttributesDTO	Service to fetch all regular CASA accounts, TD accounts and external accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	fetchLNInformation		
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	primaryAccountHolderIDMaintenance	Maintain primary Account Holder ID for Loan Account. @param accountId,newpartyId	Maintain primary Account Holder ID for Loan Account. @param accountId,newpartyId
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	primaryAccountHolderIDMaintenance	Maintain primary Account Holder ID for Loan Account. @param accountId,newpartyId	If there is a change in primary ownership of account then it will update the Loan module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	primaryAccountHolderIDMaintenance	Maintain primary Account Holder ID for Loan Account. @param accountId,newpartyId	If there is a change in primary ownership of account then it will update the TD module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	primaryAccountHolderIDMaintenance	Maintain primary Account Holder ID for Loan Account. @param accountId,newpartyId	If there is a change in primary ownership of account then it will update the DDA module for the change in ownership.  @param accountId @param newPrimaryPartyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	modifyStatusOfLoanAccounts	@param loanAccounts @param LoanAccountStatusType	Method to update the status of all the accounts held by the customer to Regular, in case customer is removed as Blacklisted. @param partyId
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	modifyStatusOfLoanAccounts	@param loanAccounts @param LoanAccountStatusType	Method to update the status of all the accounts held by the customer to Total Block, in case customer is Blacklisted. @param partyId
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	inquireLoanBalance	Method to fetch Outstanding, RPA and unclear balances for a given loan account @param accountId @throws FatalException	Method to fetch Outstanding, RPA and unclear balances for a given loan account @param accountId @throws FatalException
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	inquireLoanAccountInformation	This method is responsible for fetching the loan account details @param sessionContext, accountId @return LoanAccountUnifiedInquiryResponse	This method is responsible for fetching the loan account details @param sessionContext, accountId @return LoanAccountUnifiedInquiryResponse
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	inquireLoanAccountInformation	This method is responsible for fetching the loan account details @param sessionContext, accountId @return LoanAccountUnifiedInquiryResponse	
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	inquireLoanDisbursementDetails	This method is responsible for fetching the loan disbursement details @param sessionContext, accountId @return LoanAccountUnifiedInquiryResponse	This method is responsible for fetching the loan disbursement details @param sessionContext, accountId @return LoanDisbursementLogHistoryResponse





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	getOfferId	This method is responsible for fetching offerId given loan accountId @param sessionContext, accountId @return LoanAccountUnifiedInquiryResponse	This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray
ICustomerLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerLoansAdapter	fetchLoanAccountPreferences		This method returns the Loan account preferences details for given accountId. @param accountId @return
ICaseCreationWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICaseCreationWorkflowAdapter	getUserRole		getUserRole
ICaseCreationWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICaseCreationWorkflowAdapter	getCaseTypeCd		getCaseTypeCd

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSimulationAdapter	simulateAndFetchFutureBalanceDetails	This method invokes Full PayOff Simulation Process.  It interacts with Simulation module re-assemble the loan balance information provided by Simulation process. @param accountId @param expectedClosureDate	This method will be called to populate pay off details, which includes future arrears Unbilled amount.  @param applicationContext @param processDate @param simulationType @param partialPayOffAmount TODO @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session ticket and user id @fcb.param MI,String,accountId,1..16, The loan account number. @fcb.param MI,Date,effectiveDate, , The date on which the simulation is going to be process. @return LoanEventSimulationInquiryResponse   @throws FatalException  
ILoanSimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSimulationAdapter	simulateHardshipRelief	The method will simulate loan account lifecycle till its maturity for a given hardship relief. @param accountId @param maturityDate @param loanHardshipReliefDetailsDTO @return	The method interacts with simulation module and simulate loan account lifecycle till its maturity for a given hardship relief. @param sessionContext @param loanHardshipReliefDetailsDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSimulationAdapter	simulateRollover	This method will simulate a future dated rollover and return a simulated schedule post future dated rollover. @param accountId @param loanRolloverInstructionDTO @return	This method is used for schedule simulation during loan rollover.  <b>Algorithm</b> <ol> <li>If rollover date equals the process date, then call {@link LoanCustomerInitiatedRolloverApplication#simulateRollover(SessionContext, LoanRolloverInstructionDTO, Date)} and get the {@link LoanScheduleOutputResponse}. <li>If rollover date is after the process date, then call {@link LoanSimulationAdapter#simulateRollover} and get the {@link LoanScheduleOutputResponse} and return the response. <li>If rollover date is before the process date, then populate the backdated transaction dto by calling {@link LoanScheduleSimulationApplicationService#populateBackdatedTransactionDTO(ApplicationContext, LoanRolloverInstructionDTO)} and then {@link BackdatedRolloverApplicationService#performRolloverOnBackValueDate(SessionContext, LoanBackdatedTransactionDTO)} and get the {@link LoanBackdatedTransactionResponse} which is set in the {@link LoanScheduleOutputResponse}. </ol> @param sessionContext @param loanRolloverInstructionDTO @return @throws FatalException
ILoanSimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSimulationAdapter	simulateFutureDateSplit	This method will simulate a future dated split and return a simulated schedule post future dated split. @param accountId @param loanRepaymentScheduleConfigurationDTO @return	This method is used as a part of Loan Origination Structure Solution process to define summary as well as blown up schedule before opening a loan account. This method internally calls validateLoanScheduleConfigurationDetails method which validates the loan schedule configuration details. Also it calls defineSchedule method which creates the dummy account with given details and generates default schedule based on the given schedule configuration details. The method populateLoanScheduleConfigurationDetails also calls

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>generateRepaymentRecords which generates the blown up schedule based on the summary schedule defined in defineSchedule call.&lt;br&gt;</p> <p><b>Algorithm</b></p> <ul style="list-style-type: none"> <li>Get the parent account id for split from the <a href="#">LoanRepaymentScheduleConfigurationDTO#getParentAccountIdForSplit()</a>. <li>If the parent account id is not null then check whether arrear exists for the account. <li>Create a dummy response <a href="#">LoanDummyScheduleResponse</a> by calling <a href="#">LoanScheduleSimulationApplicationService#validateLoanScheduleConfigurationDetails(SessionContext, LoanRepaymentScheduleConfigurationDTO, String, AccountScheduleAttributesDTO)</a> and get the next repayment date from the dummy response. <li>Set <a href="#">AccountScheduleAttributesDTO#setAccountNominatedForBundlePeriodicFee(boolean)</a> from the <a href="#">LoanRepaymentScheduleConfigurationDTO</a>. <li>If <a href="#">LoanRepaymentScheduleConfigurationDTO#getScheduleGenerationDate()</a> is after process date, then call <a href="#">LoanSimulationAdapter#simulateFutureDateSplit(parentAccountIdForSplit, loanRepaymentScheduleConfigurationDTO)</a> and return the response. <li>If <a href="#">UserModifiedStageParameterDTO</a> is not null then : <ul style="list-style-type: none"> <li>Set the <a href="#">LoanRepaymentScheduleConfigurationDTO#getRepaymentDay()</a> as <a href="#">UserModifiedStageParameterDTO#getFirstPrincipalRepaymentDate()</a> <li>Get the <a href="#">LoanRepaymentStageDTO</a> from the <a href="#">LoanDummyScheduleResponse#getRepaymentStageD</a></li> </li></ul> </li> </li></li></li></li></li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>TOs()). &lt;li&gt;Call the {@link LoanScheduleSimulationApplicationService#checkNextRepaymentDate(UserModifiedStageParameterDTO, LoanRepaymentStageDTO[])}. &lt;li&gt;Derive the {@link LoanRepaymentStageResponse} by calling {@link LoanScheduleSimulationApplicationService#updateStageOnParameterChange(SessionContext, String, Integer, Date, Integer, UserModifiedStageParameterDTO, LoanRepaymentStageDTO[], SortedMap, List, LoanRepaymentScheduleConfigurationDTO)} &lt;li&gt;Get the array of {@link LoanRepaymentStageDTO} from the {@link LoanDummyScheduleResponse#getRepaymentStageDTOs()} and set the repaymentStageDTO derived from {@link LoanRepaymentStageResponse}. &lt;li&gt;Get the loanScheduleResponse {@link LoanScheduleResponse} by calling {@link LoanScheduleSimulationApplicationService#updateScheduleOnStageChange(SessionContext, String, Integer, Integer, Date, LoanRepaymentStageDTO[], SortedMap, List, LoanRepaymentScheduleConfigurationDTO)}. &lt;li&gt;Set the loanRepaymentStageDTO from the loanScheduleResponse. &lt;li&gt;If loanRepaymentStageDTO is not null, then &lt;ul&gt; &lt;li&gt;Load the {@link PrincipalScheduleTemporalPeriod} with the loanRepaymentStageDTO. &lt;li&gt;Derive the nextRepaymentDate by looping through the array of LoanRepaymentStageDTO. &lt;/ul&gt; &lt;/ul&gt; &lt;li&gt;Add bundleId and bundleReferenceId to the {@link LoanTransactionAttributesService}. &lt;li&gt;Derive the {@link LoanBlowupScheduleResponse} dummyBlowupScheduleResponse by calling {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanScheduleSimulationApplicationService#generateRepaymentRecords(SessionContext, String, Integer, Date, BigDecimal, LoanRepaymentStageDTO[], SortedMap, List, LoanRepaymentScheduleConfigurationDTO, AccountScheduleAttributesDTO)}. &lt;li&gt;Fill the final response {@link LoanScheduleOutputResponse} by:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;response.setSessionId(dummySummaryScheduleResponse.getSessionId());</li> <li>&lt;li&gt;response.setRepaymentStageDTOs(loanRepaymentStageDTO);</li> <li>&lt;li&gt;response.setRepaymentRecordsDTOs(dummyBlowupScheduleResponse.getRepaymentRecordsDTOs());</li> <li>&lt;li&gt;response.setNextRepaymentDate(nextRepaymentDate);</li> <li>&lt;li&gt;response.setYearBaseType(dummySummaryScheduleResponse.getYearBaseType()); &lt;/ul&gt;</li> <li>&lt;li&gt;Delete dummySummaryScheduleResponse by calling {@link LoanScheduleSimulationApplicationService#deleteTemporalData(LoanDummyScheduleResponse)}.</li> <li>&lt;li&gt;Fill {@link TransactionStatus} and set it in the response. &lt;li&gt;Return the response to the caller. &lt;/ol&gt;</li> </ul> <p>@fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,LoanRepaymentScheduleConfigurationDTO,loanRepaymentScheduleConfigurationDTO, , This field captures the LoanRepaymentScheduleConfigurationDTO.&lt;br&gt; LoanRepaymentScheduleConfigurationDTO will have one-to-many relationship with</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanRepaymentStageDTO.&lt;br&gt; @fcb.param MI,UserModifiedStageParameterDTO,changedStageParameterDTO, , This field captures the modified stage details if any. @return {@link com.ofss.fc.app.loan.schedule.response.LoanScheduleOutputResponse} The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;li&gt; {@link com.ofss.fc.common.CommonErrorConstants#MID_ELN_YR_RAT_INV_CHNG_INPT_999}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.CommonErrorConstants#MID_ELN_PROD_EXPIRED}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_ELN_EFF_DAT_HOLIDAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_ELN_EFF_DAT_BEYON_30_CAL_DAY}&lt;/li&gt; &lt;/ul&gt; @fcb.task LN501 @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentCommissionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentCommissionAdapter	fetchCommissionDetails	This service will fetch all the commission details for the given CommissionRequestDTO Depending upon commissionEventType provided in the commissionRequestDTO corresponding Lending/DD/OD/TD will be used to decide the commission for a rebate request agentArrangementLinkge will contain the rebated amount details @param sessionContext @param commissionRequestDTOD @return @throws FatalException	This method will fetch the commission details for the broker or introducer sourced account This service will validate the inputs and pass it to External Commission interface. The service will return the commission details. @param commissionRequestDTO
IAgentCommissionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentCommissionAdapter	fetchCommissionDetails	This service will fetch all the commission details for the given CommissionRequestDTO Depending upon commissionEventType provided in the commissionRequestDTO corresponding Lending/DD/OD/TD will be used to decide the commission for a rebate request agentArrangementLinkge will contain the rebated amount details @param sessionContext @param commissionRequestDTOD @return @throws FatalException	This service returns the rebate benefit margin value based on the upfront and trail rebate rate @param commissionRequestDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentCommissionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentCommissionAdapter	fetchRebateBenefitMargin	Adapter method to fetch rebate benefit margin based on the rebate rate provided for   upfront as well as trail commission   @param sessionContext @param commissionRequestDTOD @return AgentArrangementLinkageDTO @throws FatalException	This service returns the rebate benefit margin value based on the   upfront and trail rebate rate   @param commissionRequestDTO
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	This method performs online settlement for the given amount.  In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement	This method performs online settlement for the given amount.  In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @param shortNarration The short description for transaction @param sourceEventId The source event for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doOnlineSettlement(java.lang.String, java.lang.String, com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, com.ofss.fc.datatype.Money, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doOnlineSettlement(java.lang.String, java.lang.String, com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, com.ofss.fc.datatype.Money, java.lang.String)</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting  @param productCode The product id  @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc  @param amount The amount for which settlement needs to be done  @param settlementValueDate Value date for settlement execution  @param narration The description for online transactions  @param messageType {@link MessageType} used for settlement  @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting  @param productCode The product id  @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc  @param amount The amount for which settlement needs to be done  @param settlementValueDate Value date for settlement execution  @param narration The description for online transactions  @param messageType {@link MessageType} used for settlement  @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method calls settlement module for payout.  @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction  @return {@link TransactionDetailResponse} Object containing redemption details, transaction details after settlement</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventId The event code which will be raised for accounting @param productId The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method call the settlement module for saving the excess amount details and account logging. @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>The method would act as a wrapper class to call {@link ITDSettlementAdapter}.&lt;br&gt;The method fills the DTO sent to settlement as argument @param accountId Account id used for payin @param depositId Deposit id used for payin @param payinAmount Amount used for payin @param settlementValueDate Value date for payin settlement @param componentPreferenceDTO {@link ComponentPreferenceDTO} Contains details related to settlement mode @param narration Narration used to perform payin @return {@link PayinSettlementResponse} Result of payin settlement</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	This method performs online settlement for the given amount.  In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement	This method performs online settlement for the given amount.  In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @param shortNarration The short description for transaction @param sourceEventId The source event for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doOnlineSettlement(java.lang.String, java.lang.String, com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, com.ofss.fc.datatype.Money, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doOnlineSettlement(java.lang.String, java.lang.String, com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, com.ofss.fc.datatype.Money, java.lang.String)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting  @param productCode The product id  @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc  @param amount The amount for which settlement needs to be done  @param settlementValueDate Value date for settlement execution  @param narration The description for online transactions  @param messageType {@link MessageType} used for settlement  @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method calls settlement module for payout.  @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction  @return {@link TransactionDetailResponse} Object containing redemption details, transaction details after settlement</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventId The event code which will be raised for accounting  @productCode The product id  @componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc  @amount The amount for which settlement needs to be done  @settlementValueDate Value date for settlement execution  @narration The description for online transactions  @messageType {@link MessageType} used for settlement  @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>This method call the settlement module for saving the excess amount details and account logging. @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doOnlineSettlement	<p>This method performs online settlement for the given amount.&lt;br&gt; In online transactions, there will be main amount and fee amount. For the settlement of these amounts, it needs to be called separately.&lt;br&gt; @param eventCode The event code which will be raised for accounting  @param productCode The product id  @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc  @param amount The amount for which settlement needs to be done  @param settlementValueDate Value date for settlement execution  @param narration The description for online transactions  @param messageType {@link MessageType} used for settlement  @return {@link SettlementExecutionLogDTO} Object containing details of settlement</p>	<p>The method would act as a wrapper class to call {@link ITDSettlementAdapter}.&lt;br&gt;The method fills the DTO sent to settlement as argument  @param accountId Account id used for payin  @param depositId Deposit id used for payin  @param payinAmount Amount used for payin  @param settlementValueDate Value date for payin settlement  @param componentPreferenceDTO {@link ComponentPreferenceDTO} Contains details related to settlement mode  @param narration Narration used to perform payin  @return {@link PayinSettlementResponse} Result of payin settlement</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method invokes executePreferenceOnline method of SettlementExecutionService for Batch Transactions. It reads the Settlement Response and updates the Transaction Detail @param sessionContext @param transactionDetail @param depositDTO @return @throws FatalException @throws FCRException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doSettlement(java.lang.String, java.lang.String, java.lang.String, java.lang.String, com.ofss.fc.datatype.Money, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @param shortNarration The short description for transaction @param sourceEventId The source event for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doSettlement(java.lang.String, java.lang.String, java.lang.String, java.lang.String, com.ofss.fc.datatype.Money, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	<p>This method performs settlement for the given amount as per predefined schedule&lt;br&gt; This settlement happens in batch process.&lt;br&gt; @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException</p>	<p>This method calls settlement module for payout. @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction @return {@link TransactionDetailResponse} Object containing redemption details, transaction details after settlement</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method invokes executePreferenceOnline method of SettlementExecutionService for Batch Transactions. It reads the Settlement Response and updates the Transaction Detail @param sessionContext @param transactionDetail @param depositDTO @return @throws FatalException @throws FCRException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doSettlement(java.lang.String, java.lang.String, java.lang.String, java.lang.String, com.ofss.fc.datatype.Money, java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	doSettleme nt	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException	This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @param shortNarration The short description for transaction @param sourceEventId The source event for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doSettlement(java.lang.String, java.lang.String, java.lang.String, java.lang.String, com.ofss.fc.datatype.Money, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	doSettlement	<p>This method performs settlement for the given amount as per predefined schedule&lt;br&gt; This settlement happens in batch process.&lt;br&gt; @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException</p>	<p>This method calls settlement module for payout. @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction @return {@link TransactionDetailResponse} Object containing redemption details, transaction details after settlement</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	addComponentPreference	This method calls ComponentPreferenceApplicationService#addComponentPreference to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException	This method processes the component preference DTO and saves the component details @param componentPreferenceDTO @param payoutInstruction @param productId @param insertFlag @throws FatalException
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	addComponentPreference	This method calls ComponentPreferenceApplicationService#addComponentPreference to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException	This method calls ComponentPreferenceApplicationService#addComponentPreference to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#addComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, java.lang.String, com.ofss.fc.enumeration.settlement.ComponentSplitType[], java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	addComponentPreference	This method calls <code>ComponentPreferenceApplicationService#addComponentPreference</code> to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws <code>com.ofss.fc.infra.exception.FatalException</code>	This method calls <code>ComponentPreferenceApplicationService#addComponentPreference</code> to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws <code>com.ofss.fc.infra.exception.FatalException</code> @see <code>com.ofss.fc.app.adapter.td.ITDSettlementAdapter#addComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, java.lang.String, com.ofss.fc.enumeration.settlement.ComponentSplitType[], java.lang.String)</code>
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	carryForwardComponentPreference	This method calls <code>ComponentPreferenceApplicationService#addComponentPreference</code> to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws <code>com.ofss.fc.infra.exception.FatalException</code>	This method processes the component preference DTO and saves the component details @param componentPreferenceDTO @param payoutInstruction @param productId @param insertFlag @throws <code>FatalException</code>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method is responsible for the processing of future dated redemption Action  This method would be invoked to process actions during beginning of day for the account whose action details is passed in input.   The main job of This method is to accept the inputs and invoke <a href="#">TermDepositRedemptionApplicationService#inquireAndDoRedemption()</a> with the same.   After the <a href="#">TDActionLogger#logTDFutureDatedRedemptionAction()</a> is called to log the future dated redemption event in TD.</br> Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey <a href="#">DepositBasedActionKey</a> Object containing account and deposit id @param actionRequest <a href="#">TDActionRequest</a> Object containing td action to be performed @param dueDate Due date for the action
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method closes an active payout instruction. @param payoutInstruction @return IPayoutInstruction
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method deletes an active payout instruction. @param payoutInstruction @return IPayoutInstruction
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method reopens a closed payout instruction operation. @param payoutInstruction @return IPayoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	fetchComp onentPrefer ence		This method carries forward payout instructions from old Deposit to new Deposit. This method will be invoked at Maturity in the following cases <ul><li>Deposit is Blocked and Default Product level instruction is Reinvest</li> <li>Deposit is Liened</li> <li>Payout Instructions have been maintained to reinvest Part or Full amount to Existing Account or New Account</li> <ul> It will also be invoked at Interest Payout or at Premature Redemption if Amount is being used to open a new Deposit and at Deposit Post Maturity Renewal. The service will not carry forward Redemption instructions only the below instruction types will be carried forward only if one of the settlement modes is TDA or NTDA, i.e. part or full amount is being used to Reinvest to Existing Account or New Account. <ul><li>Interest Payout Instructions</li> <li>Principal Payout Instructions</li> <li>PrincipalInterest Payout Instructions</li> </ul> @param accountId @param depositId @param newAccId @param newDepositId @return void
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	fetchComp onentPrefer ence		This method provides the payout instruction category set in Maturity for Term deposit. @param accountId @param depositId @return String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method performs settlement for the given amount as per predefined schedule  This settlement happens in batch process.  @param arrangementID Term deposit and account id with "#" separator @param settlementPreferenceID Indicates existing component preference @param eventCode The event code which will be raised for accounting @param productCode The product id @param amountForSettlement The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for transaction @param shortNarration The short description for transaction @param sourceEventId The source event for transaction @return {@link SettlementExecutionLogDTO} Object containing details of settlement @throws FCRException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#doSettlement(java.lang.String, java.lang.String, java.lang.String, java.lang.String, com.ofss.fc.datatype.Money, java.lang.String)
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method calls ComponentPreferenceApplicationService#addComponentPreference to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#fetchComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	fetchComp onentPrefer ence		This method calls ComponentPreferenceApplicationService#addCompo nentPreference to add the component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} object which needs to be added @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#fetc hComponentPreference(com.ofss.fc.app.settlement.dto .ComponentPreferenceDTO)
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	fetchComp onentPrefer ence		This represents the fetchPayoutInstruction method of the entity. @param MI,com.ofss.fc.app.context.SessionContext,sessionCon text, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI,PayoutInstructionDTO,payoutInstructionDTO,pay outInstructionDTO @return {@link PayoutInstructionResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This represents the reopenPayoutInstruction method of the entity. @param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI,PayoutInstructionDTO,payoutInstructionDTO, ,payoutInstructionDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method generates the {@link PayoutInstructionInquiryDTO} from {@link PayoutInstruction} @param payoutInstruction @return PayoutInstructionInquiryDTO
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchComponentPreference		This method will check for particular deposit if there is any out fund flow scenario in its payout instructions @param deposit @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	updateComponentPreference	This method calls ComponentPreferenceApplicationService#updateComponentPreference to update existing component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be updated @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException	This method processes the component preference DTO and saves the component details @param componentPreferenceDTO @param payoutInstruction @param productId @param insertFlag @throws FatalException
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	updateComponentPreference	This method calls ComponentPreferenceApplicationService#updateComponentPreference to update existing component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be updated @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException	This method calls ComponentPreferenceApplicationService#updateComponentPreference to update existing component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} object which needs to be updated @param eventCode The event code which will be raised for accounting @param allowedSplitTypes Array of allowed {@link ComponentSplitType} object @param productCode The product id @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#updateComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO, java.lang.String, com.ofss.fc.enumeration.settlement.ComponentSplitType[], java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method is responsible for the processing of future dated redemption Action  This method would be invoked to process actions during beginning of day for the account whose action details is passed in input.   The main job of this method is to accept the inputs and invoke {@link TermDepositRedemptionApplicationService#inquireAndDoRedemption()} with the same.   After the TDActionLogger {@link TDActionLogger#logTDFutureDatedRedemptionAction()} is called to log the future dated redemption event in TD.</br> Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} Object containing account and deposit id @param actionRequest {@link TDActionRequest} Object containing td action to be performed @param dueDate Due date for the action
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method closes an active payout instruction. @param payoutInstruction @return IPayoutInstruction
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method deletes an active payout instruction. @param payoutInstruction @return IPayoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	closeComp onentPrefer ence	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#closeComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO)
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	closeComp onentPrefer ence	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method closes all the Settlement Instructions for the array of ComponentPreference ID @param accountId @param depositId @param componentPreferenceIDs
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	closeComp onentPrefer ence	This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed	This method closes the Principal, PrincipalInterest and Interest Instructions @param maturityProcessInformation
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	reopenCom ponentPrefer ence	This method opens the provided component preference which was closed earlier. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be reopened	This method reopens a closed payout instruction operation. @param payoutInstruction @return IPayoutInstruction
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	reopenCom ponentPrefer ence	This method opens the provided component preference which was closed earlier. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be reopened	This method opens the provided component preference which was closed earlier. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be reopened @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#reopenComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	reopenComponentPreference	This method opens the provided component preference which was closed earlier. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be reopened	This represents the reopenPayoutInstruction method of the entity. @param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI,PayoutInstructionDTO,payoutInstructionDTO, ,payoutInstructionDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	reversePreferenceOnline	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumber Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDTO} Object containing settlement execution response and accounting amount tag details	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumber Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDTO} Object containing settlement execution response and accounting amount tag details @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#reversePreferenceOnline(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	reversePref erenceOnlin e	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumbe r Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDT O} Object containing settlement execution response and accounting amount tag details	This method reverses the paying transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction. @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @param subSequenceNo TODO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	reversePref erenceOnlin e	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumbe r Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDT O} Object containing settlement execution response and accounting amount tag details	This method reverses the interest payout transaction. @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction. @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	reversePreferenceOnline	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumber Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDTO} Object containing settlement execution response and accounting amount tag details	This method reverses the redemption transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \td\ITDSettlementAdap ter	reversePref erenceOnlin e	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionreferenceNumbe r Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDT O} Object containing settlement execution response and accounting amount tag details	This method reverses the contract modification transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	reversePreferenceOnline	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose.  @param productId {@link String} It holds the productId under which the deposit was opened @param originalTransactionReferenceNumber Transaction reference number used to fetch execution details of transaction @param narration The description for online transactions @param eventId The eventId for online transactions @return {@link TermDepositSettlementResponseDTO} Object containing settlement execution response and accounting amount tag details	If a deposit is created with wrong information, it needs to be reversed.  On reversal its various amounts need to be settled.  The below method calls settlement module to serve the purpose. The method also validates the response from settlement.  @param productId It holds the productId under which the deposit was opened @param originalTransactionReferenceNumber Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param narration The description for online transactions @param eventId The eventId for online transactions.  @return {@link TermDepositSettlementResponseDTO} Object containing settlement execution response and accounting amount tag details
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchPreferenceExecutionDTOs	This method fetches the Settlement Execution details for particular Transaction @param transactionReferenceNumber Transaction reference number used to fetch execution details of transaction @return Array of {@link PreferenceExecutionDTO} objects containing settlement execution details	This method fetches the Settlement Execution details for particular Transaction @param transactionReferenceNumber Transaction reference number used to fetch execution details of transaction @return Array of {@link PreferenceExecutionDTO} objects containing settlement execution details @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#fetchPreferenceExecutionDTOs(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	fetchPreferenceExecutionDTOs	This method fetches the Settlement Execution details for particular Transaction @param transactionReferenceNumber Transaction reference number used to fetch execution details of transaction @return Array of {@link PreferenceExecutionDTO} objects containing settlement execution details	This method fetches the Settlement Details for given Transaction reference number. @param tdSettlementAdapter @param transactionReferenceNumber @return SettlementModeDTO[]
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method is responsible for the processing of future dated redemption Action  This method would be invoked to process actions during beginning of day for the account whose action details is passed in input.   The main job ofThis method is to accept the inputs and invoke {@link TermDepositRedemptionApplicationService#inquireAndDoRedemption()} with the same.   After the TDActionLogger {@link TDActionLogger#logTDFutureDatedRedemptionAction()} is called to log the future dated redemption event in TD.</br> Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} Object containing account and deposit id @param actionRequest {@link TDActionRequest} Object containing td action to be performed @param dueDate Due date for the action
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method closes an active payout instruction. @param payoutInstruction @return IPayoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method deletes an active payout instruction. @param payoutInstruction @return IPayoutInstruction
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method closes the provided component preference. @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which needs to be closed @see com.ofss.fc.app.adapter.td.ITDSettlementAdapter#closeComponentPreference(com.ofss.fc.app.settlement.dto.ComponentPreferenceDTO)
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method closes all the Settlement Instructions for the array of ComponentPreference ID @param accountId @param depositId @param componentPreferenceIDs
ITDSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSettlementAdapter	closeComponentPreference		This method closes the Principal, PrincipalInterest and Interest Instructions @param maturityProcessInformation
ICollectionAccountWriteOffAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionAccountWriteOffAdapter	performAccountFullWriteOff		
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	startJobs		This method will initiate the Extraction algorithm for the batch category by calling method startETLJobs  of ETLJobRequestor. @param l_parser

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	startJobs		
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	startJobs		
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	startJobs		This method will initiate the Extraction. @param l_parser
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	restartJobs		
IBatchDataIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchDataIntegrationAdapter	restartJobs		This method will initiate the Extraction. @param l_parser

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method performs redemption. @param redemptionStrategy {@link RedemptionStrategy} Strategy for redemption based on redemptionType @param deposit {@link Deposit} Object having deposit details @param redemptionType {@link RedemptionType} Type of redemption @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction @param raiseAccountingFlag Flag if true raise accounting event @return {@link RedemptionResponseDetailsDTO} Object containing details of redemption after actual redemption</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} Object containing details of accounting event @return {@link TransactionStatus} Status of raising accounting event transaction @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.td.dto.accounting.TDAccountingEventDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method accepts the list of TDAccountingEventDTO and calls raiseAccountingEvent with members of the list. @param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(java.util.List)</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method pays out the Outstanding Interest. It receives the amount which is payable in the {@link OutstandingInterestAmountBusinessPolicyDTO.interestPayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Finally the method persists the Transaction Detail and updates the Deposit @param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method pays out the Online Principal and Interest as a combined Amount. It receives the amount which is payable in the {@link OutstandingPrincipalInterestBusinessPolicyDTO.netAmountPayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Any unsettled or remaining amount is put in the Principal Buckets and not the Interest buckets. Finally the method persists the Transaction Detail and updates the Deposit</p> <p>@param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method pays out the Online Principal. It receives the amount which is payable in the {@link OutstandingPrincipalBusinessPolicyDTO.principalBalancePayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Finally the method persists the Transaction Detail and updates the Deposit @param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method logs the Accounting Event with the Accounting Module. This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table <b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param transactionDetail @param tdAccountingEventDTO @param accountingEvent Constant from TDAccountingEventConstants @param amountTag Constant from TDAccountingAmountTagConstants @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>Based on the flag isRedemptionRenewalforUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePricipalDue and UncollectedPricipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The RenewalDTO coming as input parameter will have all values meant for renewal of the account. But, it will have older depositId @param sessionContext This parameter captures the basic information about the user of the system. @param renewalDTO {@link RenewalDTO} contains details of the fields need to renew a deposit. @return TransactionStatus {@link TransactionStatus} holds the status of the transaction. @throws fatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#renewDeposit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.renewal.RenewalDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method settles the Excess Principal amount back to the customer, This method is called when the principal amount is decreased as a part of contract modification @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method call the settlement module for saving the excess amount details and account logging. @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method updates the branch for the account and all of its regular deposits. This method when called via branch process. Then the deposit array and the termDepositAccount Entities are null. @param accountId @param newBranchId @param branchTransferDate @param Deposit [] {@link Deposit} @param termDepositAccount {@link TermDepositAccount} The method first validates the branch transferDate, The branch transfer date should be Greater than or equal to posting date and then it checks if the account is eligible for branch transfer&lt;br&gt; if the account is eligible for the transfer then the execution of the branch transfer happens by following the below steps &lt;ul&gt; &lt;li&gt;The list of deposits for the account is fetched by calling {@link DepositService} : fetchAllDeposits() method&lt;/li&gt; &lt;li&gt;Iterate through the deposit list and pick those deposits that has deposit status as regular&lt;/li&gt; &lt;ol&gt; &lt;li&gt;update the branchId of the deposit by calling updateDepositBranch() method&lt;/li&gt; &lt;li&gt;raise an Accounting Event by calling {@link TDAccountingAdapter} : raiseAccountingEvent&lt;/li&gt; &lt;li&gt;intimate the branch id change to Interest by calling {@link TDInterestAdapter} : updateDepositBranchInInterestPlanItem&lt;/li&gt; &lt;li&gt;intimate the branch change to tax by calling {@link TDTaxationAdapter} : updateDepositBrachInTax() &lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;finally after looping through the deposit list and updating the deposits&lt;br&gt; raise an alert by logging a nonfinancial Event by calling {@link TDEventLogService} :logNonFinancialEvent()&lt;/li&gt; &lt;li&gt;update the branch for the account by calling the updateAccountBranch&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method does the Interest processing for Deposit and internally calls . @param interestInstruction @param interestPayoutInformation @param deposit</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method logs the Accounting Event with the Accounting Module. This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;br&gt; @param transactionDetail @param tdAccountingEventDTO @param accountingEvent Constant from TDAccountingEventConstants @param amountTag Constant from TDAccountingAmountTagConstants @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>The method logs the Accounting Event with the Accounting Module. @param tdAccountingEventDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @<b>param</b> termDepositAccountingEventDTO {<b>@link</b> TDAccountingEventDTO} object containing details of accounting event @<b>return</b> {<b>@link</b> TransactionStatus} status of raising accounting event transaction</p>	<p>This method invokes call to Raise Accounting. There are two methods to raise Accounting Event, the method to call is decided by this method. &lt;p&gt; If both Deposit and Product Level Maturity Instructions are not present then {<b>@link</b> raiseAccountingWithNoInstruction} is invoked &lt;/p&gt; &lt;p&gt; If Maturity Instruction are present either at Product or Deposit level then {<b>@link</b> raiseAccountingWithInstruction} is invoked &lt;/p&gt; @<b>param</b> maturityProcessInformation @<b>param</b> principalInstruction @<b>param</b> principalInterestInstruction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This is meant to change the depositStatus of a deposit to regular from fundAwaited, If the isSuccess comes true else close the deposit. @param accountId {@link String} holds the account Id under which the deposit is present. @param depositId {@link Integer} holds the deposit id of the deposit. @param isSuccess {@link Boolean} the flag from settlement, which will tell us to change the deposit status. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTO {@link TDAccountingEventDTO} object containing details of accounting event @return {@link TransactionStatus} status of raising accounting event transaction</p>	<p>This method is used to payin deposit.&lt;br&gt; Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method performs redemption. @param redemptionStrategy {@link RedemptionStrategy} Strategy for redemption based on redemptionType @param deposit {@link Deposit} Object having deposit details @param redemptionType {@link RedemptionType} Type of redemption @param redemptionTransactionDTO {@link RedemptionTransactionDTO} Object containing information of redemption transaction @param raiseAccountingFlag Flag if true raise accounting event @return {@link RedemptionResponseDetailsDTO} Object containing details of redemption after actual redemption</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p><b>@param</b> termDepositAccountingEventDTOs list of <a href="#">TDAccountingEventDTO</a> Objects containing details of accounting event <b>@return</b> <a href="#">TransactionStatus</a> Status of raising accounting events transaction</p>	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p><b>@param</b> termDepositAccountingEventDTO <a href="#">TDAccountingEventDTO</a> Object containing details of accounting event <b>@return</b> <a href="#">TransactionStatus</a> Status of raising accounting event transaction <b>@see</b> com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.td.dto.accounting.TDAccountingEventDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method accepts the list of TDAccountingEventDTO and calls raiseAccountingEvent with members of the list.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseAccountingEvent(java.util.List)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params  termDepositAccountingEventDTOs  list of {@link TDAccountingEventDTO} Objects containing details of accounting event  @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method pays out the Outstanding Interest. It receives the amount which is payable in the {@link OutstandingInterestAmountBusinessPolicyDTO.interestPayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Finally the method persists the Transaction Detail and updates the Deposit @param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method pays out the Online Principal and Interest as a combined Amount. It receives the amount which is payable in the {@link OutstandingPrincipalInterestBusinessPolicyDTO.netAmountPayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Any unsettled or remaining amount is put in the Principal Buckets and not the Interest buckets. Finally the method persists the Transaction Detail and updates the Deposit</p> <p>@param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params  termDepositAccountingEventDTOs  list of {@link TDAccountingEventDTO} Objects containing details of accounting event  @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method pays out the Online Principal. It receives the amount which is payable in the {@link OutstandingPrincipalBusinessPolicyDTO.principalBalancePayable} It invokes online settlement of the Payout Instruction. The Response received from Settlement is used to update the {@link TransactionDetail.settlementReferenceNumber} and the Deposit Balance buckets. Finally the method persists the Transaction Detail and updates the Deposit @param transactionSubsequenceNumber @param transactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param transactionDetail @param tdAccountingEventDTO @param accountingEvent Constant from TDAccountingEventConstants @param amountTag Constant from TDAccountingAmountTagConstants @throws FatalException</p>	<p>The method logs the Accounting Event with the Accounting Module. This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param transactionDetail @param tdAccountingEventDTO @param accountingEvent Constant from TDAccountingEventConstants @param amountTag Constant from TDAccountingAmountTagConstants @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> &gt; which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>Based on the flag isRedemptionRenewalforUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePricipalDue and UncollectedPricipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The RenewalDTO coming as input parameter will have all values meant for renewal of the account. But, it will have older depositId @param sessionContext This parameter captures the basic information about the user of the system. @param renewalDTO {@link RenewalDTO} contains details of the fields need to renew a deposit. @return TransactionStatus {@link TransactionStatus} holds the status of the transaction. @throws fatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#renewDeposit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.renewal.RenewalDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method settles the Excess Principal amount back to the customer, This method is called when the principal amount is decreased as a part of contract modification @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method call the settlement module for saving the excess amount details and account logging. @param deposit {@link Deposit} Term deposit details @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param oldPrincipalAmount Principal amount of deposit before contract modification @param transactionReferenceNumber Transaction reference number which will be used for settlement @param transactionSubsequenceNumber Transaction sequence number @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params  termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method updates the branch for the account and all of its regular deposits. This method when called via branch process. Then the deposit array and the termDepositAccount Entities are null. @param accountId @param newBranchId @param branchTransferDate @param Deposit [] {@link Deposit} @param termDepositAccount {@link TermDepositAccount} The method first validates the branch transferDate, The branch transfer date should be Greater than or equal to posting date and then it checks if the account is eligible for branch transfer  if the account is eligible for the transfer then the execution of the branch transfer happens by following the below steps</p> <ul style="list-style-type: none"> <li>The list of deposits for the account is fetched by calling {@link DepositService} : fetchAllDeposits() method</li> <li>Iterate through the deposit list and pick those deposits that has deposit status as regular</li> <li>update the branchId of the deposit by calling updateDepositBranch() method</li> <li>raise an Accounting Event by calling {@link ITDAccountingAdapter} : raiseAccountingEvent</li> <li>intimate the branch id change to Interest by calling {@link TDInterestAdapter} : updateDepositBranchInInterestPlanItem</li> <li>intimate the branch change to tax by calling {@link TDTaxationAdapter} : updateDepositBrachInTax()</li> </ul> <p>finally after looping through the deposit list and updating the deposits  raise an alert by logging a nonfinancial Event by calling {@link TDEventLogService} :logNonFinancialEvent()  update the branch for the account by calling the updateAccountBranch  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params  termDepositAccountingEventDTOs  list of {@link TDAccountingEventDTO} Objects containing details of accounting event  @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method does the Interest processing for Deposit and internally calls . @param interestInstruction @param interestPayoutInformation @param deposit</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>&lt;br&gt; @param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method logs the Accounting Event with the Accounting Module. This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>&lt;br&gt; @param transactionDetail @param tdAccountingEventDTO @param accountingEvent Constant from TDAccountingEventConstants @param amountTag Constant from TDAccountingAmountTagConstants @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>The method logs the Accounting Event with the Accounting Module. @param tdAccountingEventDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p><b>@param</b> termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event <b>@return</b> {@link TransactionStatus} Status of raising accounting events transaction</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.  @params  termDepositAccountingEventDTOs  list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method invokes call to Raise Accounting. There are two methods to raise Accounting Event, the method to call is decided by this method. &lt;p&gt; If both Deposit and Product Level Maturity Instructions are not present then {@link raiseAccountingWithNoInstruction} is invoked &lt;/p&gt; &lt;p&gt; If Maturity Instruction are present either at Product or Deposit level then {@link raiseAccountingWithInstruction} is invoked &lt;/p&gt; @param maturityProcessInformation @param principalInstruction @param principalInterestInstruction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This is meant to change the depositStatus of a deposit to regular from fundAwaited, If the isSuccess comes true else close the deposit. @param accountId {@link String} holds the account Id under which the deposit is present. @param depositId {@link Integer} holds the deposit id of the deposit. @param isSuccess {@link Boolean} the flag from settlement, which will tell us to change the deposit status. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method with input list of TDAccountingEventDTO which calls raiseAccountingEvent to raise the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table</p> <p><b>&lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt;</b> which will have one to one relationship with behaviour<b>&lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt;</b> and one to many relationship with details <b>&lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;</b>.</p> <p>@param termDepositAccountingEventDTOs list of {@link TDAccountingEventDTO} Objects containing details of accounting event @return {@link TransactionStatus} Status of raising accounting events transaction</p>	<p>This method is used to payin deposit.&lt;br&gt; Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction	This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction @see com.ofss.fc.app.adapter.td.ITDAccountingAdapter#raiseReversalAccountingEvent(java.lang.Integer, java.lang.String, java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction</p>	<p>This method reverses the paying transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction. @param isAccountingReversalRequired informs that the accounting is to be raised @param taskCode Identifier of the transaction to be reversed @param subSequenceNo TODO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction</p>	<p>This method reverses the interest payout transaction. @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction. @param isAccountingReversalRequired informs that the accounting is to be raised @param taskCode Identifier of the transaction to be reversed</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction</p>	<p>This method reverses the redemption transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.&lt;br&gt; @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.&lt;br&gt; @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction</p>	<p>This method reverses the renewal transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.&lt;br&gt; @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.&lt;br&gt; @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction	This method reverses the contract modification transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountingAdapter	raiseReversalAccountingEvent	This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param sequenceNumber Event sequence no of accounting event to be reversed @param transactionReferenceNo Transaction reference no which will be used for reversed event @param originalTransactionReferenceNo Transaction reference no of the event to be reversed @return {@link TransactionStatus} Status of raising reversed accounting event transaction	This method is used for raising a reversal event for the transaction and event sequence no provided.   Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. @param internalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber of the transaction currently in execution. @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.
ITDAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountingAdapter	getNextAccountingEventSequenceNumber	This method will return next Accounting Event sequence number @param transactionRefNumber Transaction reference number for which event sequence number is required @return Next event sequence number to be used	This is meant to change the depositStatus of a deposit to regular from fundAwaited, If the isSuccess comes true else close the deposit. @param accountId {@link String} holds the account Id under which the deposit is present. @param depositId {@link Integer} holds the deposit id of the deposit. @param isSuccess {@link Boolean} the flag from settlement, which will tell us to change the deposit status. @throws FatalException
IGEFUDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUDDAAdapter	creditSalaryAccount	@param accountId @param bsbCode @param eventId @param serviceCode @param amount @return @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IGEFUDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUDDAAdapter	debitSalary Account	@param accountId @param bsbCode @param eventId @param serviceCode @param amount @return @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler
IGEFUDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUDDAAdapter	raiseAccounting	@param accountingEventAmountTagDetails DTOs @param transactionReferenceNumber @param transactionDescription @param primaryAccountId @param eventId @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler
IGEFUDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUDDAAdapter	raiseAccounting	@param accountingEventAmountTagDetails DTOs @param transactionReferenceNumber @param transactionDescription @param primaryAccountId @param eventId @throws FatalException	
IGEFUDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUDDAAdapter	raiseAccounting	@param accountingEventAmountTagDetails DTOs @param transactionReferenceNumber @param transactionDescription @param primaryAccountId @param eventId @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsContentAdapter	generateLetterOfWaiverAndNonWaiver		This method represents issuerLetterGeneration method of the entity. This method is used to generate covering letter for the parties. @param MI, com.ofss.fc.app.context.SessionContext,sessionContext ,This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI, CovenantLinkageWrapperDTO wrapperDTO @param MI, boolean flag @return IssuerLetterResponse @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section @param sessionContext @param wrapperDTO @param flag
ICollateralsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsContentAdapter	generateConditionLetterOfWaiverAndNonWaiver		
ICollateralsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsContentAdapter	generateConditionLetterOfWaiverAndNonWaiver		Generate Letter Of Waiver And NonWaiver for Conditions. Letter is generate may be more than one according partyId, partyId varying according BorrowingEntity @param covenantLinkageWrapperDTO @param flag @return @throws FatalException
ICollateralsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsContentAdapter	generateConditionLetterOfWaiverAndNonWaiver		Generate Letter Of Waiver And NonWaiver for Conditions. Letter is generate may be more than one according partyId, partyId varying according BorrowingEntity @param covenantLinkageWrapperDTO @param flag @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsContentAdapter	generateExternalTermDepositLeinAdvice		
IConsultingInitializationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\IConsultingInitializationAdapter	initialize		Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller, delayed transaction poller, FTP webservice poller. Please add more in this method. Additionally we also have an extension for consulting pollers @param sessionContext
IConsultingInitializationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\IConsultingInitializationAdapter	initialize		Initialize webservice annotation processor. This injects OWSM policy on all webservices @param sessionContext
IDDAStatementPrintAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAStatementPrintAdapter	printDDStatement		This method calls the OutboundDocumentApplicationService.generateSetOfDocument for printing. This method uses {@link DemandDepositConfiguration} for getting parameters required to print the report. @param sessionContext @param statementPrintingDTO
IDDAStatementPrintAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAStatementPrintAdapter	printDDStatement		This method is used to print DD statements. This method gets call from UI. Method needs call from UI because the request in table flx_dd_statement_gen_items won't get populated till Interaction.close() of orderDDStatements(). That method will commit the data into DB and gives call back to UI. UI will then call this method and this method call IDDAStatementPrintAdapter.printDDStatement() to actually print the statements. @param sessionContext @param ddStatementRequestDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	saveSettlementAsDraft		<p>This service is used to save the settlement details as draft, The ApplicationSettlementDTO is obtained as input. Using the same, the settlement payout instruction dto is saved as draft in settlement module - settlement reference and linkages are constructed and saved in origination. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param ApplicationSettlementDTO, Contains the settlement details for each offer in the application @return TransactionStatus @exception {@link OriginationErrorConstants#SETTLEMENT_VALIDATIONS_FAILED} : : FC_OR_SI_0021 &lt;br&gt; This exception occurs when any of the following validations fail &lt;br&gt; 1. Mandatory fields in the input ApplicationSettlementDTO are null {@link OriginationErrorConstants#OR_INV_INPUT} :FC_OR_DE_0019 &lt;br&gt; 2.No account has been nominated for fees collection.&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_NOMINATED_ACCOUNT} : FC_OR_SI_0045 &lt;br&gt; 3.If the settlement mode is 'APPLICATION', there can be only one nominated account&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_MODE_ONE_NOMINATED_ACCOUNT} : FC_OR_SI_0046&lt;br&gt; 4.No settlement instructions have been set up for the nominated account &lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_STLMNT_INS} : FC_OR_SI_0002 &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	fetchDraftSettlement		This method will fetch the settlement payout instruction dto. If settlement status = Draft, the settlement will be fetched from Draft table the Confirmed Settlement will be fetched otherwise @param payoutId @return SettlementPayoutInstructionDTO @throws FatalException
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	fetchConfirmedSettlement		This method will submit the Settlement instructions task. The lending application state transitions from SettlementInstructions captured to settlement instructions completed state, The following checks are done: 1. Accounts have been opened for all the PG's that do not require settlement 2. Settlement has been completed for all the PG's that requires settlement If the above two validations are successful, the application is marked as Processed and if the collateral perfection task , if any, has been completed, the submission is marked as Fulfilled @param SessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @param String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @throws FatalException @exception   {@link OriginationErrorConstants#SETTLEMENT_SUBMIT_FAILED} : FC_OR_SI_0001   This error occurs when user submits the settlement task and the Settlement cannot be submitted as the Contribution pay in and Settlement payout instructions for at least one account should be executed    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	fetchConfirmedSettlement		This method will fetch the settlement payout instruction dto. If settlement status = Draft, the settlement will be fetched from Draft table the Confirmed Settlement will be fetched otherwise @param payoutId @return SettlementPayoutInstructionDTO @throws FatalException
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	saveConfirmedSettlement		This service is used to save the settlement details as Confirmed. The ApplicationSettlementDTO is obtained as input. <li>If settlement exists/does not exist for the application in DRAFT mode, - The settlement instructions are Confirmed - The settlement reference in Origination are updated - Lending account is created <li>If settlement exists in CONFIRMED mode, The settlement instructions are updated- The settlement references in Origination is updated, but account creation is not called @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param ApplicationSettlementDTO, Contains the settlement details for each offer in the application @return TransactionStatus @exception {@link OriginationErrorConstants#OR_AUTO_REFI_POLICY_NOT_SETTLED} :FC_OR_AR_0005   This exception occurs when any of the Auto Refinance Policies under the Application   are not Settled and/or OFI Payout Amount is not received.     {@link OriginationErrorConstants#SETTLEMENT_REVOKE_FAILED} : FC_OR_SI_0003   This exception occurs when settlement revoke fails. Settlement can only be revoked if the Contribution pay in and (or) Settlement payout instructions for at least one account have been executed     {@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#SETTLEMENT_VALIDATIONS_FAILED} : FC_OR_SI_0021 &lt;br&gt; This exception occurs when any of the following validations fail &lt;br&gt;</p> <p>1. Mandatory fields in the input ApplicationSettlementDTO are null {@link OriginationErrorConstants#OR_INV_INPUT} :FC_OR_DE_0019 &lt;br&gt; 2.No account has been nominated for fees collection.&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_NOMINATED_ACCOUNT} : FC_OR_SI_0045 &lt;br&gt; 3.If the settlement mode is 'APPLICATION', there can be only one nominated account&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_MODE_ONE_NOMINATED_ACCOUNT} : FC_OR_SI_0046&lt;br&gt; 4.No settlement instructions have been set up for the nominated account &lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_STLMNT_INS} : FC_OR_SI_0002 &lt;br&gt; 5.The total disbursement amount is more than the approved amount for the facility &lt;br&gt; {@link OriginationErrorConstants#OR_DISBURSEMENT_AMOUNT_EXCEEDED} : FC_OR_SI_0049 &lt;br&gt; 6.The total disbursement amount is less than the approved amount for the LOAN or MOF facility &lt;br&gt; {@link OriginationErrorConstants#OR_DISBURSEMENT_AMOUNT_LESS} : FC_OR_SI_0049 &lt;br&gt; 7.The contribution amount (The fees get collected from the contribution amount or as part of disbursement)is insufficient for fees collection. &lt;br&gt; {@link OriginationErrorConstants#OR_INSUFFICIENT_CONTRIBUTION_AMOUNT} : FC_OR_SI_0051</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	updateComponentPreferenceDTOWithPartyId		This method inserts the pay out instruction in the database. It takes the pay out instructions from the {@link TDAccountConfigurationDTO} and the sets the account id into pay out instructions. @param sessionContext @param tdAccountConfigurationDTO @throws FatalException
IOriginationSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationSettlementAdapter	processSettlementInstructionNow		This service is used to save the settlement details as Confirmed. The ApplicationSettlementDTO is obtained as input. <li>If settlement exists/does not exist for the application in DRAFT mode, - The settlement instructions are Confirmed - The settlement reference in Origination are updated - Lending account is created <li>If settlement exists in CONFIRMED mode, The settlement instructions are updated- The settlement references in Origination is updated, but account creation is not called @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param ApplicationSettlementDTO, Contains the settlement details for each offer in the application @return TransactionStatus @exception {@link OriginationErrorConstants#OR_AUTO_REFI_POLICY_NOT_SETTLED} :FC_OR_AR_0005   This exception occurs when any of the Auto Refinance Policies under the Application   are not Settled and/or OFI Payout Amount is not received.     {@link OriginationErrorConstants#SETTLEMENT_REVOKE_FAILED} : FC_OR_SI_0003   This exception occurs when settlement revoke fails. Settlement can only be revoked if the Contribution pay in and (or) Settlement payout instructions for at least one account have been executed     {@link OriginationErrorConstants#SETTLEMENT_VALIDAT

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IONS_FAILED} : FC_OR_SI_0021 &lt;br&gt; This exception occurs when any of the following validations fail &lt;br&gt;</p> <p>1. Mandatory fields in the input ApplicationSettlementDTO are null {@link OriginationErrorConstants#OR_INV_INPUT} :FC_OR_DE_0019 &lt;br&gt;</p> <p>2.No account has been nominated for fees collection.&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_NOMINATED_ACCOUNT} : FC_OR_SI_0045 &lt;br&gt;</p> <p>3.If the settlement mode is 'APPLICATION', there can be only one nominated account&lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_MODE_ONE_NOMINATED_ACCOUNT} : FC_OR_SI_0046&lt;br&gt;</p> <p>4.No settlement instructions have been set up for the nominated account &lt;br&gt; {@link OriginationErrorConstants#SETTLEMENT_NO_STLMNT_INS} : FC_OR_SI_0002 &lt;br&gt;</p> <p>5.The total disbursement amount is more than the approved amount for the facility &lt;br&gt; {@link OriginationErrorConstants#OR_DISBURSEMENT_AMOUNT_EXCEEDED} : FC_OR_SI_0049 &lt;br&gt;</p> <p>6.The total disbursement amount is less than the approved amount for the LOAN or MOF facility &lt;br&gt; {@link OriginationErrorConstants#OR_DISBURSEMENT_AMOUNT_LESS} : FC_OR_SI_0049 &lt;br&gt;</p> <p>7.The contribution amount (The fees get collected from the contribution amount or as part of disbursement)is insufficient for fees collection. &lt;br&gt; {@link OriginationErrorConstants#OR_INSUFFICIENT_CONTRIBUTION_AMOUNT} : FC_OR_SI_0051</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISessionContextAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISessionContextAdapter	populateSessionContext	This method is used to populate SessionContext.  @fcb.param MI,String,channel, Indicates transaction channel.  @throws FatalException	This method is used to populate SessionContext.  @fcb.param MI,String,channel, Indicates transaction channel.  @throws FatalException
ISessionContextAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISessionContextAdapter	populateSessionUser		This method is used to populate SessionContext.  @fcb.param MI,String,channel, Indicates transaction channel.  @throws FatalException
IIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates CoverContext attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException
IIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates Context attributes for the extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates CoverContext attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates HypothecationContext attributes for the HypothecationLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates PersonalGuaranteeContext attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationCommonServiceAdapter	fetchBranchParameters	This represents the fetchBranchParameters method of the entity. @return {@link com.ofss.fc.app.commonservice.core.dto.branchparameter.BranchParametersInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method populates OfferContext attributes for the offerLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException
IIIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationCommonServiceAdapter	fetchBankParameters	This represents the fetchBankParameters method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BankParametersDTO,bankParametersDTO, ,bankParametersDTO TODO: Explain the bankParametersDTO above @return {@link com.ofss.fc.app.BankParametersDefinitionInquiryResponse.dto.bankparameter.BankParametersInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationCommonServiceAdapter	fetchBankParameters	<p>This represents the fetchBankParameters method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,BankParametersDTO,bankParametersDTO, ,bankParametersDTO TODO: Explain the bankParametersDTO above @return {@link com.ofss.fc.app.BankParametersDefinitionInquiryResponse.dto.bankparam.BankParametersInquiryResponse} @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBankParameters	<p>This represents the fetchBankParameters method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,BankParametersDTO,bankParametersDTO, ,bankParametersDTO TODO: Explain the bankParametersDTO above @return {@link com.ofss.fc.app.BankParametersDefinitionInquiryResponse.dto.bankparam.BankParametersInquiryResponse} @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will populate Context of Documaker. @param sessionContext @param applicationID @return Context @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBankParameters	<p>This represents the fetchBankParameters method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,BankParametersDTO,bankParametersDTO, ,bankParametersDTO TODO: Explain the bankParametersDTO above @return {@link com.ofss.fc.app.BankParametersDefinitionInquiryResponse.dto.bankparam.BankParametersInquiryResponse} @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will populate Context of Documaker. @param sessionContext @param applicationID @return Context @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchBankParameters	This represents the fetchBankParameters method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BankParametersDTO,bankParametersDTO, ,bankParametersDTO TODO: Explain the bankParametersDTO above @return {@link com.ofss.fc.app.BankParametersDefinitionInquiryResponse.dto.bankparam.BankParametersInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This method populates populateMortgageContext attributes for the MortgageLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return Context @throws FatalException
IntegrationCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationCommonServiceAdapter	fetchFinancialYearCode	This method gets the current financial year code @param bankCode @param branchCode @param date @return	This method return the current financial year code @param sessionContext @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardNumber	Inquire Credit Card Details for card number @param sessionContext @param cardNumber @return CreditCardDetailsDTO @throws FatalException	The method inquireCardBasicDetails is used for fetching card basic details for the given card number.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,cardNumber, ,This attribute denotes the card number which is identifier for credit card number. @return {@link CreditCardDetailsInquiryResponse} This class contains the Credit Card details. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardNumber	Inquire Credit Card Details for card number @param sessionContext @param cardNumber @return CreditCardDetailsDTO @throws FatalException	Inquire Credit Card Details for card number @param sessionContext @param cardNumber @return CreditCardDetailsDTO @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardReferenceNumber	Inquire Credit Card Details for card number @param sessionContext @param cardNumber @return CreditCardDetailsDTO @throws FatalException	Inquire Credit Card Details for card reference number @param sessionContext @param cardReferenceNumber @return CreditCardDetailsDTO @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardReferenceNumber	Inquire Credit Card Details for card reference number @param sessionContext @param cardReferenceNumber @return CreditCardDetailsDTO @throws FatalException	The method changeCardLimit is used to change the limit for the credit card account.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, ,This attribute denotes the card account number. @fcb.param MI,MoneyDTO, requestLimit, ,This attribute denotes the request limit. @return {@link TransactionStatus} This class returns the transaction status @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardReferenceNumber	Inquire Credit Card Details for card reference number @param sessionContext @param cardReferenceNumber @return CreditCardDetailsDTO @throws FatalException	The method inquireCardAccountDetails is used to fetch credit card account details for the given card account number.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, This attribute denotes the card account number. @return {@link CreditCardAccountDetailsResponse} This class contains the information related to credit card account details. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardReferenceNumber	Inquire Credit Card Details for card reference number @param sessionContext @param cardReferenceNumber @return CreditCardDetailsDTO @throws FatalException	The method inquireCardDetailsByAccountNumber is used for fetching card details for the given internal account number.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountNumber, This attribute denotes the internal account number. @return {@link CreditCardDetailsInquiryResponse} This class contains the Credit Card details. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICreditCardAdapter	inquireCreditCardDetailsForCardReferenceNumber	Inquire Credit Card Details for card reference number @param sessionContext @param cardReferenceNumber @return CreditCardDetailsDTO @throws FatalException	This method is used to fetch Credit Card Details for internal or external party. The party details and credit card reference number are fetched from MDM and using the credit card reference number, the card details are fetched from the credit card external system @param sessionContext @param partyId @param externalPartyId @return creditCardDetailsDTOs, CreditCardDetailsDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICreditCardAdapter	openCreditCardAccount	Open Create Card Account in the external system @param sessionContext @param creditCardAccountInputDTO @return CreditCardDetailsDTO @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference. After that it will save the current state of simulation. Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICreditCardAdapter	performPostOpenCreditCardAccountOperation	Perform Post Open Credit Card Account Operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardDetailsDTO @throws FatalException	The method performPostOriginateCreditCardOperation is responsible for doing post originate credit card operation.   @fcb.param MI, SessionContext, sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	amendCard Limit	Amend Card Limit @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method amendCreditCard is responsible for amend credit card, invoked from origination flow   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also interact with the external card system and amend the card limit.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	amendCard Limit	Amend Card Limit @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method changeCardLimit is used to change the limit for the credit card account.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, ,This attribute denotes the card account number. @fcb.param MI,MoneyDTO, requestLimit, ,This attribute denotes the request limit. @return {@link TransactionStatus} This class returns the transaction status @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	performPostAmendCardLimitOperation	Perform Post Amend Card Limit Operation @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method performPostAmendCreditCardOperation is responsible for doing post amend credit card operation.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	linkAddOnCardHolder	Link Add on card holder @param sessionContext @param addOnCardHolderDTO @return AddOnCardHolderDTO @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICreditCard Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICreditCardAdapter	updateBundleBenefits	Update bundle benefits is used to update the bill code in the external system @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method inquireCardAccountDetails is used to fetch credit card account details for the given card account number.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CardBundleLinkageDTO cardBundleLinkageDTO, This attribute contains account id, bundle id, bundle reference id, is bundle benefit applicable flag, bundle effective start / end date and bill code. @return {@link TransactionStatus} This class returns the transaction status @throws FatalException
IUISVCParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\uisvc\IUISVCPartyAdapter	fetchAccountsOfParty		
IUISVCParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\uisvc\IUISVCPartyAdapter	fetchAccountTypeForAccount		
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This method will be used to inquire the available overdraft limit for the account. In order to get this, call the method {@link IDDAeLCMAadapter#inquireFacility} After getting overdraft limit and utilized overdraft limit, the attributes of this class are set. grantedOverdraftLimit = facilityDetailsDTO.getOverdraftLimit(); utilizedOverdraftLimit = facilityDetailsDTO.getUtilizedOverdraftLimit(); </pre> @param balanceCurrency @param valueDate @param accountId



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This method will get the balance availability of account balances for the value date. This method if value date equals to posting date then it will call LCM to get the data of response.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This method returns the applicable AUF limit by taking into consideration the various AUF parameters in this class and the uncleared funds and the posting date passed. This method works as follows :  <pre> applicableAUFLimit = 0 if( postingDate is before and equals endDate    postingDate is after and equals startDate) &lt;i&gt; posting date must fall in between startdate and end date. return applicableAUFLimit = 0; else { minimum of (lendablePercentage unclearFunds, sanctionedLimitAmount) } </pre> @param unclearedFunds @param postingDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This will inquire the current balance availability of the given account and make future availability of the balance to be null. @param account @param effectiveDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This will inquire the future balance availability of the given account and make the current availability as zero balance. The inquiry is done by using (@link IDDLCMAdapter#getBalancesAvailabilityOfAccount) @param account @param postingDate @param processDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This method is used to calculate and return the applicable AUF limit for the account. It takes into consideration the unclear funds and the AUF Limit parameters to arrive at the applicable limit.  The method works as follows: <ul> <li>Multiply the lendablePercentage with the input.unclearFunds</li> <li>Take the least of the value calculated above and the aufLimitAmount of the history record and

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				return.</li> @param dates
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	getBalancesAvailabilityOfAccount	This @param accountId @param valueDate @return	This method is used to get the applicable overdraft limit for the record. The method works as follows:   <ul> <li>If the facilityId at history record level is null, the return 0.</li> <li>If the facilityID is not null, then call the elcm service through the adapter {<a href="#">@link IDDLCMAdapter#getBalancesAvailabilityOfAccount</a>} to get the facility details.</li> <li>Extract the value of the overdraftLimit from the response returned by the service above and return.</li> </ul> @param dates
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateELCM	Intimate ELCM with the balances. @param messageType	Intimate ELCM @param dates @param overDraftLimit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateELCM	Intimate ELCM with the balances. @param messageType	This method intimates eLCM about the balance utilization and deutilization. This method is called on every financial transaction. This method works as follows :   <pre> Derive Principal balance for setBal() method. principalBalance = UtilizedOverdraftAmount() +TODAmount() + OverlineAmount() {The Overdraft, TOD and Overline amount are extracted from the input balance map} if(principalBalance > 0) { &lt;i&gt;If it is available then pass it, else pass null.&lt;/i&gt; &lt;i&gt;In case of ODLimit utilization or Overline Utilization it will be passed to the ELCM service, If TOD is utilized, then it may or may not be derived.&lt;/i&gt; if(FCRThreadAttribute.AUTH_REQUIRED == false) { try { Call {@link IDDAELCMAdapter#setBal()} &lt;i&gt;params : Balance : principalBalance, forceAuth : 'N', AmountTag : 'PRINCIPAL'&lt;/i&gt; } catch(Override Exception e) { Call {@link IDDAELCMAdapter#setBal()} &lt;i&gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'PRINCIPAL'&lt;/i&gt; } } if(FCRThreadAttribute.AUTH_REQUIRED == true) { Call {@link IDDAELCMAdapter#setBal()} &lt;i&gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'PRINCIPAL'&lt;/i&gt; } } aufBalance = UtilizedAUFLimit() {UtilizedAUFLimit is extracted from the balance map} if(aufBalance > 0) { &lt;i&gt;No need to pass utilizationId&lt;/i&gt; Call {@link IDDAELCMAdapter#setBal()} &lt;i&gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'AUF'&lt;/i&gt; } </pre> @param accountId @param dates @param balanceMap @param utilizationId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateELCM	Intimate ELCM with the balances. @param messageType	This method intimates eLCM about the balance utilisation and deutilisation. This method is called on every financial transaction. This intimates balance category to account.  @param demandDepositAccount @param postBalanceMap @param preBalanceMap @param dates
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateELCM	Intimate ELCM with the balances. @param messageType	This method intimates eLCM about the balance utilisation and deutilisation. This method is called on every financial transaction.  @param demandDepositAccount @param postBalanceMap @param preBalanceMap @param dates
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchPartyForBorrowingEntity	@param sessionContext @param borrowingEntityCode @return	
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchPartyForBorrowingEntity	@param sessionContext @param borrowingEntityCode @return	This method validates borrowing entity id. @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchPartyForBorrowingEntity	@param sessionContext @param borrowingEntityCode @return	This method will fetch all Accounts related to Borrowing Id. @param sessionContext @param borrowingEntityCode @return accountArray @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBorrowingEntityForParty		
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBorrowingEntity		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBorrowingEntity		this will call and load borrowing entity.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBorrowingEntity		9NT1487ST-1632 : Method to set borrowing entity even when OD facility has expired @param sessionContext @param demandDepositAccount @return @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	inquireFacilityDetails		The method will check the following: <ol style="list-style-type: none"> <li>Is facilityId valid</li> <li>Is facility started</li> <li>This method validates the account currency and the facility code currency. If the account currency is not same as facility currency then we throw exception (Account currency and facility currency should be same).</li> </ol>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	inquireFacilityDetails		The method will create a new OD linkage. It does the following: <ol style="list-style-type: none"> <li>Validate business policy using business policy instance returned from {@link AccountFacilityLinkageBusinessPolicy#createODLinkagePolicyInstance()}</li> <li>If the linkage start date is equal to current process date, do linkage online by calling {@link ODLimitLinkageExecutor#linkODFacilityOnline()}</li> <li>Else log an {@link DDActionType#ODLimitLinkage} with due date as link start date.</li> <li>If the fixedRatePeriod is not null, then log a {@link DDActionType#ODFixedRatePeriodEnd} action with dueDate as the next day BOD of fixedRatePeriod end date.</li> <li>Log {@link DDActionType#ODLimitLinkageExpiry} action with dueDate as the next day of linkage period end date.</li> <li>Call {@link #create()} method to persist the linkage.</li> </ol> @param accountFacilityLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	inquireFacilityDetails		This method performs following linkage date related validations. <ol style="list-style-type: none"> <li>Offset linkage end date can not be greater than Overdraft facility expiry date of the beneficiary account. </li></ol>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	startODLinkage		This method will perform the job of linking given od facility to given demandDepositAccount on current posting date. The steps involved in the processing are as follows. <ul style="list-style-type: none"> <li>Fetch latest limits from LCM.Stamp it on account. The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account. These are parameters which are common for linkage on current date and for back date.</li> <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility. So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.</li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account. So that LCM can realign their account with proper facilities</li> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls {@link</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging.</li> <li>Now it intimates the NPA,Collection module about the linkages.</li> </ul> @param demandDepositAccount @param accountLinkage @param isODRateFixed
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	ODLinkage Expiry		This method is used by ELCM to intimate DDA about the OD Closure. The method accepts the accountId and accountFacilityLinkage.The method has to delink if facility is applicable and remove previous facility from account. @param sessionContext @param accountId @param accountFacilityLinkage @return @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	ODLinkage Expiry		This method handles the common of linkage parameters such as od facility id,od facility start date ,od expiry date,determination of shared facility, borrowing entity id,fixed rate start,end dates, available amounts of given accounts. @param demandDepositAccount @param accountLinkage @param ddLcmAdapter @param isODRateFixed @param transactionDates @param odSanctionedMoney @param effectiveLimitAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	ODLinkage Expiry		<p>This represents the closeAccountFacilityLinkage method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,AccountFacilityLinkageDTO AccountFacilityLinkageDTO. It contains following details &lt;ol&gt; &lt;li&gt;AccountFacilityLinkageKeyDTO:Key for entity AccountFacilityLinkage, contains account Id and Facility Id, using this information we can inquire about a facility linked to a account.i.e its start date , end date , fixed rate duration if applied.</p> <p>&lt;li&gt;fixedRatePeriod:Fixed rate period defines the start date end date for the period through which the account is charged a fixed rate.fixed rate period start date should always be equal to linkage period start date. &lt;li&gt;linkagePeriod:Linkage Period defines the Start Date and End Date for the facility linked to the account &lt;/ol&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	ODLinkageExpiry		<p>This represents the closeFacilityLinkageForLoanAccountOpening method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,AccountFacilityLinkageDTO AccountFacilityLinkageDTO. It contains following details &lt;ol&gt; &lt;li&gt;AccountFacilityLinkageKeyDTO:Key for entity AccountFacilityLinkage, contains account Id and Facility Id, using this information we can inquire about a facility linked to a account.i.e its start date , end date , fixed rate duration if applied.</p> <p>&lt;li&gt;fixedRatePeriod:Fixed rate period defines the start date end date for the period through which the account is charged a fixed rate.fixed rate period start date should always be equal to linkage period start date. &lt;li&gt;linkagePeriod:Linkage Period defines the Start Date and End Date for the facility linked to the account &lt;/ol&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	ODLinkageExpiry		<p>This override method will do the action. Action here is to delink the Overdraft Facility.</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	decreaseBackdatedUnclearedBalance		<p>This will trigger the decrease unclear funds back dated. @param amount @param accountId @param valueDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		This method increases the unclear balance of the account by 'amount'. It calls this.getBalances().increaseUnclearBalance (amount, dates). After that it intimates ELCM with decreased balance.SO that it can compute the modified AUF limit.{@link IDDLCMAdapter#increaseUnclearBalance} @param amount @param dates
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		This service will be used to update the CASA balances when a cheque is deposit in a DD account. When the cheque is deposited in an account, the unclear balance of the account is updated with the cheque amount. In case an AUF limit has been granted to the account, the available AUF amount is recalculated based on the updated unclear funds. As a result of this the withdrawable balance of the account increases. The account statement will reflect a credit entry for the account based on the statement parameter setup. For e.g - AUF Limit provided = 10,000 Margin = 10% Cheque Amount = 5,000 This will result in the following : Uncleared Funds = 5,000 AUF Amount = ((100-10)/100)5000 = 4500 Customer can utilize upto 4500 in addition to his current balances.The method works as follows: <ul style="list-style-type: none"> <li>Call the validation method {@link this.validateTransactionDTOForOutwardClearing} to validate the details of the input DTO.</li> <li>validate the status of the account(closed,unclaimed,writeoff) before the unclear balance of account is being updated.</li> <li>Call the private method {@link this.updateBalanceAndLogStatement} with the transactionType as Credit to update the unclear funds, log the financial statement item and log balance</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				change action.</li> </ul> @param sessionContext @param chequeDetailsDTO
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		This service will be used to reduce the Uncleared funds in an account when the cheque is reversed before the value date. The handling will be the same when the cheque is returned before the value date.   For more details , see {@link this.updateBalanceOnChequeReturnBeforeValueDate}. The method works as follows: <ul> <li>Call the validation method {@link this.validateTransactionDTOForOutwardClearing} to validate the details of the input DTO.</li> <li>Call the private method {@link this.updateBalanceAndLogStatement} with the transactionType as Debit to update the unclear funds, log the financial statement item and log balance change action.</li> </ul> @param sessionContext @param ClearingTransactionDTO @throws FatalException @deprecated

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		<p>This service will be used to update the CASA balances when a cheque is deposit in a DD account. When the cheque is deposited in an account, the unclear balance of the account is updated with the cheque amount.</p> <p>&lt;br&gt; In case an AUF limit has been granted to the account, the available AUF amount is recalculated based on the updated unclear funds. As a result of this the withdrawable balance of the account increases.</p> <p>&lt;br&gt; The account statement will reflect a credit entry for the account based on the statement parameter setup. &lt;br&gt; For e.g - &lt;br&gt; AUF Limit provided = 10,000 &lt;br&gt; Margin = 10% &lt;br&gt; Cheque Amount = 5,000 &lt;br&gt; &lt;br&gt; This will result in the following :</p> <p>&lt;br&gt; Uncleared Funds = 5,000 &lt;br&gt; AUF Amount = ((100-10)/100)5000 = 4500 &lt;br&gt; Customer can utilize upto 4500 in addition to his current balances.&lt;br&gt;</p> <pre>The method works as follows: &lt;ul&gt; &lt;li&gt;&lt;li&gt;Call the validation method {@link this.validateTransactionDTOForOutwardClearing} to validate the details of the input DTO.&lt;/li&gt; &lt;li&gt;&lt;li&gt;Call the private method {@link this.updateBalanceAndLogStatement} with the transactionType as Credit to update the unclear funds, log the financial statement item and log balance change action.&lt;/li&gt; &lt;/li&gt; &lt;/li&gt; &lt;/ul&gt; At the time of logging statements it uses the flag of pseudo entry. If that column is marked as true then that particular entry will not appear in the statement inquiry. At the time of logging statements it uses the flag of pseudo entry. If that column is marked as true then that particular entry will not appear in the statement inquiry. </pre> <p>@param clearingTransactionDTO @param isPseudoEntry @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		This method intimates LCM about the increase in unclear funds. <ul style="list-style-type: none"> <li>First it gives call to LCM for getting the new sanctioned money after given unclear funds deposited.</li> <li>Then it gives call to {@link DDAELCMHelper#prepareAndStampNewAvailableLimit(DemandDepositAccount, Money)} to stamp new available amount after this transaction is done.</li> <li>then it intimates LCM about increase in the amount for processing.</li> <li>This pseudo call is made first because after LCM realises the money it starts realign with other buckets as per sequence. So in that flow limit available money at LCM side may get changed.</li> </ul> @param money @param accountId @param chequeDepositDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseUnclearBalance		This method is used to update the unclear funds. This method is called when some cheque is deposited or may be any other financial transaction occurred which led to the increase or decrease in unclear funds of the account. @param accountId @param amount @param dates @param transactionType @param isLCMToBeIntimated
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	increaseBackdatedUnclearBalance		This will trigger the increase unclear funds back dated to LCM. @param amount @param accountId @param valueDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	decreaseUnclearBalance		This method decreases the unclear balance of the account by 'amount'. It calls this.getBalances().decreaseUnclearBalance(amount,dates).After that it intimates ELCM with decreased balance.SO that it can compute the modified AUF limit.{@link IDDLCMAdapter#decreaseUnclearBalance} @param amount @param dates

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	decreaseUnclearBalance		This method is used to update the unclear funds. This method is called when some cheque is deposited or may be any other financial transaction occurred which led to the increase or decrease in unclear funds of the account. @param accountId @param amount @param dates @param transactionType @param isLCMToBeIntimated
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	resetAUFLimit	This method will reset the AUF limit when a account is marked as weak. @param accountId	This method would intimate prevent against unclear funds on the account @param sessionContext @param accountWIDetailDTO @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	resetAUFLimit	This method will reset the AUF limit when a account is marked as weak. @param accountId	This method will reset the AUF limit when a account is marked as weak. @param accountId
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	resetAUFLimit	This method will reset the AUF limit when a account is marked as weak. @param accountId	
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	resetAUFLimit	This method will reset the AUF limit when a account is marked as weak. @param accountId	
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		This will create a borrowing entity in LCM.This to be called after linking of any facility.And when account is going in negative without having any facility.i.e. using TOD.And stamp that borrowing id on account.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		@param debitResponse @param ddaelcmHelper @param preArrearRunningBalance @param postArrearRunningBalance @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				calling doSanctityCheck counter inProgressCount is set to 0 so that if any other transaction which will come after that will get inProgressCount as 0 because hibernate load this class from cache. There is possibility to get counter as 0. @param amountToCredit @param referenceNumber @param dates @param financialEventType @param messageType @param eventId @param serviceId @param transactionAttributes @param subsequenceNumber @param nsfTreatment @return
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		This method completely reverses the effects of a credit which has taken place on the account at an earlier point in time. It uses the originalReferenceNumber to get the details of the original credit. TODO: Detailed steps to be written here. @param originalReferenceNumber @param originalCreditResponse @param financialEventType @param referenceNumber @param messageType @param eventId @param serviceId @return
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		This will create a borrowing entity id.And stamp it on account. @param accountId @return @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		This method merges borrowing entity details into account. The details include borrowing entity id and borrowing entity start date. If borrowing entity id is null then it stamps the existing id into that. If date is null or date is less than current posting date then it replaces that date with borrowing entity start date. @param demandDepositAccount @param accountLinkage



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		<p>This is a common method used to update funds to a demand deposit account during a backdated transaction. The service is used to update funds both during a credit and a debit transaction. When a backdated transaction occurs, the funds have to be updated from the value date of the transaction upto the current process date. This is required in order to recompute the interest for the account. The demand deposit account contains a EffectiveBalance Pair of the current and future balances. The past balances of the account are available in the BalanceHistoryItem. Hence, when a backdated transaction takes place, the records of the BalanceHistoryItem will be updated starting from the value date of the transaction. Finally, the current balance of the account will be updated. The method works as follows:</p> <ul style="list-style-type: none"> <li>Loads the demand deposit account by calling {@link DemandDepositAccountRepository#load} using accountTransaction.accountId.</li> <li>Validate the status of the account by calling the service {@link DemandDepositAccount#validateStatus} to verify whether funds can be credited to the account or not.</li> <li>Retrieve the OD limit start date and AUF start date from demand deposit account (DemandDepositAccount#getBalances#getBalanceLimits#getOverdraftLimit/#getAufLimit#startDate).</li> <li>Call the service {@link DemandDepositAccount#resetEffectiveBackDate} with the transaction value date to update the value of the attribute effectiveBackDate at the account level.</li> <li>transactionAmounts = {@link TransactionAmounts#prepareTransactionAmounts()}.</li> <li>Load the details of the balance history item from the value date of the transaction by calling the</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>service {@link BalanceHistoryItemRepository#loadHistoryForBackdatedPeriod} with the value date as the value date of the transaction.</p> <ul style="list-style-type: none"> <li>Open a loop for the list of records returned by the above service.</li> <li>If the effective date of the Balance history Item record &lt; value date of the transaction, it means there is no row present in BalanceHistoryItem as of the value date. A row needs to be created in the BalanceHistoryItem with the effective date as the value date of the transaction. Prepare a balance history Item record with the effective date as the value date of the transaction, and the rest of the details same as the loop record and use for the below processing.</li> <li>When creating new balance history item, use the OD start date and AUF start date retrieved above</li> <li>If the value of the input.transactionType = Credit, call the service {@link BalanceHistoryItem#credit} to update the credit amount in the balance history. ElseIf the value of input.transactionType = Debit, call the service {@link BalanceHistoryItem#debit} to update the debit amount in the balance history. Pass the transaction value date as the effective date of the loop record.</li> <li>Call the service {@link BalanceHistoryItemRepository#saveOrUpdate} with the details of the transaction response returned by the service above.</li> <li>Reset the value date to the posting date in the attribute TransactionDates. If the value of the input.transactionType = Credit, call the service {@link DemandDepositAccount#credit} ElseIf the value of the input.transactionType = Debit, call the service {@link DemandDepositAccount#debit}</li> <li>Update the instance of the DemandDepositAccount by calling the update() method of the repository {@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DemandDepositAccountRepository#update}&lt;/li&gt;  &lt;li&gt;Construct and log the financial statement item  &lt;li&gt;Invoke  StatementItemService.addDDFinancialStatementItem()  . The value of the transactionType should be set  according to the input parameter  input.TransactionType in the financialStatementItem.  &lt;/ul&gt; @param accountTransaction @param  transactionType @param creditResponse @param  debitResponse</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		<p>This method is used to link the AUF Facility. @param  accountId @param facilityId @param limitAmount  @param startDate @param endDate</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	createBorrowingEntity		<p>This method works in this flow. &lt;ol&gt; &lt;li&gt;If  borrowing entity id in DemandDepositAccount is null  then in that case it creates Borrowing entity id for  given account and stamps it on account. It ensures  account can use limits and collaterals with that  borrowing entity id.&lt;/li&gt; &lt;li&gt;If start date of  borrowing entity is more than current date due to  borrowing entity created after cutoff in that case start  date is replaced with current date.&lt;/li&gt; &lt;li&gt;Then it  sets limit on account.LCM has to ensure to call CASA  before aligning their funds to ensure proper limit  intimation.&lt;/li&gt; &lt;li&gt;If temporary excess is null on  account then it creates them, then it stamps the  sanctioned limit, available limit.&lt;/li&gt; &lt;li&gt;If start date  of facility is not equal to current date and it is in batch  mode then it is case of backdated TE linkage due to  holiday processing. In this case it asks pricing for  backdated linkage of stage. {@link  IDDA PricingAdapter#persistAndLogRateChangeBack  dated(String, Date, CASAInterestTypes,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>boolean))&lt;/li&gt; &lt;li&gt;If start date of facility equals to current then it asks pricing for current dated stage creation. {@link IDDApricingAdapter#persistAndLogRateChange(String, Date, CASAInterestTypes, boolean)}&lt;/li&gt; &lt;li&gt;It first now aligns current state of account by using {@link DemandDepositAccount#reapportionFunds(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)} for non batched linkage and {@link DemandDepositAccount#reapportionFundsForAmountChange(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)} for batched linkages.&lt;/li&gt; &lt;li&gt;Now for history items in range it aligns them by using {@link this#mergeSanctionAmountForHistoryItemsInRange(Date, DemandDepositAccount, Money, ApplicationContext, TransactionAttributes, TransactionDates, com.ofss.fc.enumeration.lcm.limits.FacilityType)}&lt;/li&gt; &lt;li&gt;Lastly it calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of linkage.&lt;/li&gt; &lt;/ol&gt; @param accountId @param facilityId @param limitAmount @param startDate @param endDate</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateLCMForBackdated		<p>This method intimate LCM for back value dated transactions. @param postBalanceMap @param preBalanceMap @param vdProcessingRequestDTO @param accountId @param dates @param messageType @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	intimateLCMForBackdated		This method will handle the LCM requirements which will take original and modified balance history items. at that point of time it will intimate the LCM for records having any of the quadrants -ve value. @param accountId @param valueDate @param balanceHistoryItemsOriginal @param balanceHistoryItemsModified @param dates @param preBalanceMap @param postBalanceMap @return
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	intimateELCMForReverse		This method intimates eLCM about the balance utilisation and deutilisation. This method is called on every financial transaction.  @param demandDepositAccount @param postBalanceMap @param preBalanceMap @param dates
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	updateLimitExpiryDate	This method will be called from DDAHardShipReliefService if hardShipReliefType is 07 Change Term 07 then will call the LimitsAccountApplicationService to change the term by extending limit_expiry_date. @param odFacilityId @param reliefExpiryDate	This method will be called from hardShipReliefExecution if hardShipReliefType is "Change Term" then it will call LCM adapter for HardShip Relief action for changing term. It will call the LimitsAccountApplicationService to change the term by extending limit_expiry_date. @param applicationContext @param accountId @param odFacilityId @param reliefExpiryDate @throws FatalException
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	decreaseBackdatedUnclearFundsForPeriod	this will decrease the unclear funds of account for given period.This operation is required for cheque dishonors after the value date. @param accountId @param convertMoneyDTOFromMoney @param startDate @param endDate	
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	inquireStageOfAccount	This will inquire the stage of account. @param accountId @return	This method will stamp the stage value at the account and persist it into the DB. @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchVDUtilDetailsForFacilityNereastToEffectiveDate	<p>This method will inquire the limit allowed and limit utilised for a sub-line for the due date. The due date can be a back date. </p> @param facilityId @param dueDate @return @throws FatalException	
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchVDUtilDetailsForFacilityNereastToEffectiveDate	<p>This method will inquire the limit allowed and limit utilised for a sub-line for the due date. The due date can be a back date. </p> @param facilityId @param dueDate @return @throws FatalException	<p>This method will fetch the utilisation details from ELCM for the input sub-line on the due date. </p> @param facilityId @param dueDate @return
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchVDUtilDetailsForFacilityNereastToEffectiveDate	<p>This method will inquire the limit allowed and limit utilised for a sub-line for the due date. The due date can be a back date. </p> @param facilityId @param dueDate @return @throws FatalException	This will inquire the future balance availability of the given account and make the current availability as zero balance. The inquiry is done by using (@link IDDLCMAdapter#getBalancesAvailabilityOfAccount) @param account @param postingDate @param processDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchVDUtilDetailsForFacilityNereastToEffectiveDate	<p>This method will inquire the limit allowed and limit utilised for a sub-line for the due date. The due date can be a back date. </p> @param facilityId @param dueDate @return @throws FatalException	This inquiry is done for the balance history for the given value date. The method used in the processing is {@link IDDLCMAdapter#fetchVDUtilDetailsForFacilityNereastToEffectiveDate(String, Date)} @param account @param effectiveDate
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	calculateSanctionedAmountOfAUFFacility	This will calculate the new sanctioned amount from LCM. @param accountId @param increasedAmount @return	This will stamp the new sanctioned amount from LCM and stamp it at account level. @param amount @param account .getAccountId()
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	validateProductGroup	This will validate the product group of the account with the facility to which it will be linked . @return	This is a public method exposed to the domain service to perform business validations for account facility linkage maintenance. The method internally calls the private methods to process the validations.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	validateProductGroup	This will validate the product group of the account with the facility to which it will be linked . @return	This method does all the validation that happens in LCM when facility is linked. This validation required only for future facility start date.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchEffectiveLimitOffFacility	This will fetch the sanctioned limit of facility. @param facilityId @return	This method will perform the job of linking given od facility to given demandDepositAccount on current posting date. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOffFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.</li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper facilities</li> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls {@link DemandDepositUtil#handleBalanceChangeAction(De

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				mandDepositAccount, Date, TransactionAttributes)) for handling action part of logging.</li> <li>Now it intimates the NPA,Collection module about the linkages.</li> </ul> @param demandDepositAccount @param accountLinkage @param isODRateFixed
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchEffectiveLimitOfFacility	This will fetch the sanctioned limit of facility. @param facilityId @return	This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>Now it finds out the list of {@link BalanceHistoryItem} and updates them with sanctioned OD limit amount and available amount on that date by using {@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)} <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>this limit is used for reapportion.</p> <ul style="list-style-type: none"> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account. So that LCM can realign their account with proper facilities</li> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls <code>{@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)}</code> for handling action part of logging.</li> <li>Now it intimates the NPA, Collection module about the linkages.</li> </ul> <p>@param demandDepositAccount @param accountLinkage @param isODRateFixed</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBalanceDetails		<pre> This method is called while Change OverDraft to TermLoan processing. This method performs following operations. <ul style="list-style-type: none"> <li>Prepares <code>{@link FeeDetailsDTO}</code> from <code>{@link WrappedPriceFeatureDTO}</code> and levy fees by calling <code>{@link FeeHelper#levyFee()}</code>.</li> <li>Calls <code>{@link DemandDepositAccountService#chargeEconomicCostAndRaiseAccounting()}</code> to levy Economic Cost.</li> <li>Calls <code>{@link AccountFacilityLinkageApplicationService#closeAccountFacilityLinkage()}</code> to de-link OD Facility.</li> <li>Call <code>{@link DDAInquiryApplicationService#fetchBalanceDetails()}</code> to fetch and return balances.</li> </ul> @author sambitc @param sessionContext @param accountId @param pricefeatureDTO @return BalanceInquiryDetailsDTO @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBalanceDetails		This method provides the facility to modify the attributes associated with auf limit definition. At the time of account opening, AUF limit is always zero. This method is used for AUFLimit creation also, since by default at the account opening, AUFLimit will be zero.  Call {@link AUFLimitAssembler#toDomainObject()} Call {@link DemandDepositAccountService#modifyAufLimit()} @param aufLimitDto @param sessionContext
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBalanceDetails		This method will perform the job of linking given od facility to given demandDepositAccount on current posting date. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.</li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account.So

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>that LCM can realign their account with proper facilities</p> <ul style="list-style-type: none"> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls <code>{@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)}</code> for handling action part of logging.</li> <li>Now it intimates the NPA,Collection module about the linkages.</li> </ul> <p>@param demandDepositAccount @param accountLinkage @param isODRateFixed</p>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	fetchBalanceDetails		<p>This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows.</p> <ul style="list-style-type: none"> <li>Fetch latest limits from LCM. Stamp it on account. The method which used for this is <code>{@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)}</code> here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date, od end date on account. These are parameters which are common for linkage on current date and for back date.</li> <li>Now it finds out the list of <code>{@link BalanceHistoryItem}</code> and updates them with sanctioned OD limit amount and available amount on that date by using <code>{@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)}</code></li> <li>After that it fetches the sanctioned limit from facility by using <code>{@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}</code>. The sanctioned limit can not be directly used for</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				align because there can be other accounts connected to same facility. So it fetches the effective limit of facility by using {@link IDDLCAAdapter#fetchBalanceDetails(String)}. Now this limit is used for reapportion. </li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not. </li> <li>It intimates LCM for linkage of facility and account. So that LCM can realign their account with proper facilities </li> <li>It calls reapportion for aligning the funds with CASA and LCM. </li> <li>It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging. </li> <li>Now it intimates the NPA, Collection module about the linkages. </li> </ul> @param demandDepositAccount @param accountLinkage @param isODRateFixed
IDDLCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCAAdapter	fetchBalanceDetails		This method will calculate overall utilized OD amount by account which are sharing the facility. @param odFacilityId @param transactionDates
IDDLCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCAAdapter	inquireAUFFacilityDetails		@param accountId @param demandDepositAccount
IDDLCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCAAdapter	intimateAccountStatus	This method would be called when the account status is closed, writeoff, unclaimed, it intimates lcm using the facility id, borrowing entity id @param accountId @param facilityId @param borrowingEntityId	This method would be called when the account status is closed, writeoff, unclaimed, it intimates lcm using the facility id, borrowing entity id @param accountId @param facilityId @param borrowingEntityId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	intimateAccountStatus	This method would be called when the account status is closed,writeoff,unclaimed,it intimates lcm using the facility id,borrowing entity id @param accountId @param facilityId @param borrowingEntityId	This method would intimate LCM for any OD facility associated with the account @param account @param adapter
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	intimateAccountStatus	This method would be called when the account status is closed,writeoff,unclaimed,it intimates lcm using the facility id,borrowing entity id @param accountId @param facilityId @param borrowingEntityId	This method would intimate LCM for any TE facility associated with the account @param account @param adapter
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	intimateAccountStatus	This method would be called when the account status is closed,writeoff,unclaimed,it intimates lcm using the facility id,borrowing entity id @param accountId @param facilityId @param borrowingEntityId	This method would intimate LCM for any AUF facility associated with the account @param account @param adapter
IDDLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAadapter	startBackdatedODLinkage	This method will intimate back dated OD linkage to LCM for processing. @param accountId @param facilityId @param beginDate	This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is (@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)) here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>Now it finds out the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>list of {@link BalanceHistoryItem} and updates them with sanctioned OD limit amount and available amount on that date by using {@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)} &lt;li&gt;After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.&lt;/li&gt; &lt;li&gt;Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.&lt;/li&gt; &lt;li&gt;It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper facilities&lt;/li&gt; &lt;li&gt;It calls reapportion for aligning the funds with CASA and LCM.&lt;/li&gt; &lt;li&gt;It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging.&lt;/li&gt; &lt;li&gt;Now it intimates the NPA,Collection module about the linkages.&lt;/li&gt; &lt;/ul&gt; @param demandDepositAccount @param accountLinkage @param isODRateFixed</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	updateFacilityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	This method will update the facility linkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDLCMAdapter	updateFacilityLinkage	<p>This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	updateFacilityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	<pre>This method will update the facility linkage details </pre> @param amendConfigurationDTO
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	calculateSanctionedAmountInBackdatedUnclearChange	This will calculate return the map of date and sanctioned amount from the date of cheque deposited till process date. @param accountId @param increasedAmount @return	<ul style="list-style-type: none"> <li>This method derives sanctioned and applicable amount for each value date in range for given unclear amount. The logic works as follows</li> <li>This calls LCM with value date and gets a map of applicable amount for each day by using .</li> <li>It derives the applicable list of {@link BalanceHistoryItem} till today.</li> <li>Now for each item it tries to find sanctioned limit from LCM and stamp it on that item. If the limit is not found then it uses limit of immediately previous day.</li> <li>Now for each item, it derives the effective limit of each day and stamp it on {@link BalanceHistoryItem} derived</li> </ul> <p>@param amount @param accountId @param valueDate @param accountTransaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimatePreventExcess		This method would set the prevent excess flag on the account @param sessionContext @param accountWIDetailDTO
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimatePreventExcess		
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateRemovalOfPreventExcess		This method would remove the prevent excess flag on the account @param sessionContext @param accountWIDetailDTO @throws FatalException
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	intimateRemovalOfPreventExcess		This method will actually reset the prevent excess account flag after all the reason codes are removed from the account. @param accountId
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	validateIfAccountIsMaintainedForFeeAtFacilityLevel		
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	validateIfAccountIsMaintainedForFeeAtFacilityLevel		This method populates overridable errors. <ol> <li>Account is nominated for Bundle maintenance fee(BMF). {@link DemandDepositAccountErrorConstant#OB_ACCOUNT_NOMINATED_FOR_BUNDLE_FEE}. </li> Account is nominated for facility level fees.{@link DemandDepositAccountErrorConstant#FC_DDA_ACCOUNT_IS_NOMI_AT_FAC_LVL_FOR_FEE} </ol> @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	delinkTEAndAUFOnClosure		
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	delinkTEAndAUFOnClosure		This method is used to do the Action such as delinking the facility.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	delinkTEAndAUFOnClosure		This override method will do the action. Action here is to delink the TE Facility and Intimate about the same to LCM.
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	expireHardshipAndChargeFee		This intimates Hardship relief to LCM module. @param demandDepositAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	closeFacilityOnODToLoan		<pre> This method is called while Change OverDraft to TermLoan processing. This method performs following operations. &lt;ul&gt; &lt;li&gt;Prepares {link FeeDetailsDTO}&amp;nbsp; from {link WrappedPriceFeatureDTO} and levy&lt;strong&gt; fees&lt;/strong&gt; by calling {link FeeHelper#levyFee()}.&lt;/li&gt; &lt;li&gt;Calls {link DemandDepositAccountService#chargeEconomicCostAndRaiseAccounting()} to levy &lt;strong&gt;Economic Cost&lt;/strong&gt;.&amp;nbsp;&lt;/li&gt; &lt;li&gt;Calls {link AccountFacilityLinkageApplicationService#closeAccountFacilityLinkage()} to de-link &lt;strong&gt;OD Facility.&lt;/strong&gt;&lt;/li&gt; &lt;li&gt;Call&amp;nbsp; {link DDAInquiryApplicationService#fetchBalanceDetails()} to fetch and return balances.&lt;br /&gt; &lt;br /&gt; &lt;/li&gt; &lt;/ul&gt; @author sambitc @param sessionContext @param accountId @param pricefeatureDTO @return BalanceInquiryDetailsDTO @throws FatalException </pre>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	closeFacilityOnODToLoan		<pre> Closes facility as will so that the same should not be used for further linkage. @param facilityId </pre>
IDDLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDLCMAdapter	inquireFacilitySummary	<pre> This method will fetch the facility details against the provided filter&lt;br/&gt; {link FacilityApplicationService} @param filter @return {link FacilityInquiryResponse} </pre>	<pre> Return true is more than one CASA account is linked to the credit facility&lt;br/&gt; @param sessionContext {link SessionContext} @param accountServicingFeeRequestDTO {link DDAccountConfigurationFeeRequestDTO} @return boolean @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	computeAndReturnTax	<p>This will be used to compute the tax for an arrangement number.&lt;br&gt; This will be called by the interest module.&lt;br&gt; @param taxComputationRequest @return @throws FatalException</p>	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt; This method will internally delegate tax processing call to computeAndReturnTax method of {@link TaxManagementService}.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt; This method will wrap above details inside {@link TaxProcessingResult} and return it to the caller.&lt;br&gt; Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new {@link DepositInformation} object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new {@link AccountInformation} object.&lt;br&gt; Populate {@link ProductClass} in it. Also put {@link DepositInformation} object object created in Step 1 in {@link AccountInformation#depositList}.&lt;/li&gt; &lt;li&gt;Create new {@link TaxComputationRequest} object.&lt;br&gt; Populate partyId in it. Also put {@link AccountInformation} created in Step 2 in {@link TaxComputationRequest#accountList}.&lt;/li&gt; &lt;li&gt;If (isBackDatedProcessing == true) then &lt;ol&gt; &lt;li&gt;Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Above method will return {@link TaxComputationResponse} object.&lt;/li&gt; &lt;li&gt;Create new {@link TaxProcessingResult} object populate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				information in it using {@link TaxComputationResponse} object obtained in Step 6.</li> <li>Return {@link TaxProcessingResult} object to caller.</li> </ol> @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {@link TaxComputationResult} object containing details of tax computed along with pre and post tax Interest amounts.
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	computeAndReturnTax	This will be used to compute the tax for an arrangement number.  This will be called by the interest module.  @param taxComputationRequest @return @throws FatalException	
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	computeAndReturnTax	This will be used to compute the tax for an arrangement number.  This will be called by the interest module.  @param taxComputationRequest @return @throws FatalException	This method is called to retrieve the tax computed for an arrangement   which in case of CASA is an account number and in case of TD it is combination of account number#deposit number.  This method will follow following steps to compute   The interest module will trigger a call to the TDS module for Tax computation whenever needed.  The Tax deduction can be done at different stages of interest processing like accrual, compounding and capitalization.  1. Load tax Parameters 2. If the LCY is different from the ACY, TDS module converts the interest amount from the ACY to LCY.  3. TDS module checks if Tax waiver is applicable for this account/deposit.  4.If Yes, TDS module checks if any of the following conditions are met, a) If the 'Exempt entire amount' flag is checked. b) If interest amount [in LCY] is less than equal to the threshold amount specified for the waiver or   5. If yes, TDS module intimates Interest module that Tax is not applicable for the account/deposit.   6.If no, TDS

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>module checks if any Tax exemption criterion has been maintained for the <math>\cap_{\gamma}^{\text{J}}</math> customer type <math>\cap_{\gamma}^{\text{J}}</math> and <math>\cap_{\gamma}^{\text{J}}</math> customer class <math>\cap_{\gamma}^{\text{J}}</math> of the customer holding this account/deposit.&lt;br&gt; 7.TDS module evaluates the Tax exemption based on the interest amount (in LCY), customer type, customer class and age of the customer to whom this account/deposit belongs. (In case of joint accounts, the TDS module will evaluate all the Tax exemption criteria individually for each joint account/deposit holder and the least of all exemption amounts will be used for Tax processing).&lt;br&gt; 8.If the interest amount (in LCY) is less than the exemption amount then the TDS module intimates the interest module that the account/deposit is Tax exempt.&lt;br&gt; 9.If the interest amount (in LCY) exceeds the exemption limit, the TDS module will make a call to the Pricing module (Fees) for Tax amount computation. TDS module will send the interest amount (in LCY), Account holder details (Like customer type, customer class, age; these details are used as facts for evaluating PPC<math>\cap_{\gamma}^{\text{J}}</math>s).&lt;br&gt; 10.The Pricing module will do the following on receiving the Tax computation request &lt;br&gt; &lt;li&gt;1. Evaluate the PPC<math>\cap_{\gamma}^{\text{J}}</math>s based on the customer facts passed and resolve the Base Price Definition (PD) chart/s. (The PPC<math>\cap_{\gamma}^{\text{J}}</math>s may be maintained for the product under which the account/deposit is opened or may be defined at the bank level. PPC<math>\cap_{\gamma}^{\text{J}}</math>s defined at the bank level will be applicable for all the products in the bank). &lt;li&gt;2. Retrieve the TCY code from the base PD . &lt;li&gt;3. If the TCY is different from LCY, It converts the interest amount from LCY to TCY. &lt;li&gt;4. Compute the base tax amount using the tax rate maintained in the Base PD/s. &lt;li&gt;5. Check by if there is any additional PD/s defined in the Base PD/<math>\cap_{\gamma}^{\text{J}}</math>s. &lt;li&gt;6. If Yes,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>check if any additional tax/es are applicable to this account/deposit by checking if the interest amount exceeds the exemption limit (in TCY) defined in the Additional Pd/∩∟s. &lt;li&gt;7. If yes, compute the Additional Tax/es using the Tax rate maintained in the Additional PD∩∟s. &lt;li&gt;8. Calculate the ∩∟ Total Tax amount∩∟ a. If Additional Tax/es are applicable the total Tax amount = Base Tax/es + Additional Tax/es. b. If Additional Tax/es are not applicable the total Tax amount = Bases Tax/es. &lt;li&gt;9. Pricing module checks if any more Base PD∩∟s are applicable to this account/deposit If yes it repaeats steps 2 through 9, else proceeds to step 10. &lt;li&gt;10. Pricing module will compute the Sum of all ∩∟ Total Tax amounts∩∟ as calculated in step 8 during each iteration and returns it to the TDS module along with the ∩∟ Tax type∩∟ breakup details (for inquiry purposes).&lt;li&gt;&lt;br&gt; 11.On receiving the TDS amount the TDS module will A. Apply rounding rules defined for Tax B. Log event for accounting entries for TDS in TCY. For a detailed description of the accounting entries refer section 10.1.1.2. (For a detailed explanation on the recovery of this Tax amount refer section ∩∟ Tax Recovery∩∟). C. Convert tax amount from TCY to ACY if needed. D. Return the Tax amount in ACY to the Interest module &lt;br&gt; @param SessionContext @param TaxComputationRequestDTO</p>
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	simulateTax	This method will be used to get the amount which is going to be deduct for that income when the actual tax computation is done.  @param taxSimulationRequest @return @throws FatalException	This method is used to call Tax Module for simulating tax on input of adjustment amount. Based on the inputs TaxSimulationRequest is formed and a call to TaxManagementService is done which in turn returns the tax amount. @param accountId @param partyId @param adjustmentAmount @param productClass @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	simulateTax	This method will be used to get the amount which is going to be deduct for that income when the actual tax computation is done.  @param taxSimulationRequest @return @throws FatalException	
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	simulateTax	This method will be used to get the amount which is going to be deduct for that income when the actual tax computation is done.  @param taxSimulationRequest @return @throws FatalException	
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	simulateTax	This method will be used to get the amount which is going to be deduct for that income when the actual tax computation is done.  @param taxSimulationRequest @return @throws FatalException	This method will be used to just know the simulated tax amount for an arrangement number by passing the following input. <li>TaxSimulationRequest</li> This method will not store any computation information and will not pass accounting entries .  This will be called at the time of <li>Interest adjustment</li> <li>Premature redemption</li> @param SessionContext @param TaxSimulationRequestDTO
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackdatedTaxProcessing	This method is used to do the tax computation at the time of backdated processing. @param backdatedTaxProcessingRequest @return @throws FatalException	This method is used to perform tax processing for income that is already registered with Taxation module. This method will internally delegate tax processing call to computeAndReturnTax method of {@link TaxManagementService}.  For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.  Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax. This method will wrap above details inside {@link TaxProcessingResult} and return it to the caller.  Following is the algorithm for this method:  <ol> <li>Create new {@link DepositInformation} object.  Populate eventId, arrangementNumber, valueDate in it.</li> <li>Create

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>new {@link AccountInformation} object.&lt;br&gt; Populate {@link ProductClass} in it. Also put {@link DepositInformation} object object created in Step 1 in {@link AccountInformation#depositList}.</p> <p>&lt;li&gt;Create new {@link TaxComputationRequest} object.&lt;br&gt; Populate partyId in it. Also put {@link AccountInformation} created in Step 2 in {@link TaxComputationRequest#accountList}.</p> <p>&lt;li&gt;If (isBackDatedProcessing == true) then &lt;ol&gt; &lt;li&gt;Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;</p> <p>&lt;/li&gt; &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;</p> <p>&lt;/li&gt; &lt;li&gt;Above method will return {@link TaxComputationResponse} object.&lt;/li&gt; &lt;li&gt;Create new {@link TaxProcessingResult} object populate information in it using {@link TaxComputationResponse} object obtained in Step 6.&lt;/li&gt; &lt;li&gt;Return {@link TaxProcessingResult} object to caller.&lt;/li&gt; &lt;/ol&gt; @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {@link TaxComputationResult} object containing details of tax computed along with pre and post tax Interest amounts.</p>
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackDatedTaxProcessing	This method is used to do the tax computation at the time of backdated processing. @param backdatedTaxProcessingRequest @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackdatedTaxProcessing	This method is used to do the tax computation at the time of backdated processing. @param backdatedTaxProcessingRequest @return @throws FatalException	This method is called to do the backdated tax processing for an arrangement   which in case of CASA is an account number and in case of TD it is combination of account number#deposit number. This method will be called by passing the parameters {@link BackdatedTaxProcessingRequest} and will return {@link BackdatedTaxComputationResponse}.  Following is the algorithm   <li>Load {@link TaxParameters} by passing bank code and module code.</li> <li>Call {@link PartyAccountRelationshipService#fetchAllPartiesOfAccount} to fetch the parties which are linked to that account.</li> <li>Call {@link PartyApplicationService#fetchPartyBasicDetailsInBatchMode} to fetch the party information.</li> <li>Call {@link RuleExecutionApplicationService#executeLatestRuleForGivenDate} by passing the facts to fetch the exemption limit.</li> <li>Check waiver maintained for that period , If waiver maintained for that period tax will be waived. If not call {@link TaxProcessingService#getTaxOnEvent(PricingEventContext, PricingContractContext, String, String[], Date)}</li> <li>Apply rounding method on the computed amount.</li> <li>Make a call {@link TaxRecoveryService#recoverTax(TaxRecoveryAttributes, String, String, TaxJournal, BackdatedTaxProcessingRequest)} to recover tax.</li> <li>If Recovery account maintained for an arrangement number tax amount will be recovered from the account account maintained else tax will be recovered from the income.</li> <li>Once recovered will make a call {@link TaxComputationStrategy#recordJournal(PriceFeature,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				TaxJournalAttributes, TaxRecoveryAttributes, BackdatedTaxProcessingRequest)) to record the journal tax code wise.</li> <li>Make a call to update the tax statistics by passing taxed income and computed amount.</li> <li>Will return computed details as {@link BackdatedTaxComputationResponse}.</li> @param sessionContext @param backdatedTaxProcessingRequestDTO @return TransactionStatus @throws FatalException
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackDateTaxProcessing	This method is used to do the tax computation at the time of backdated processing. @param backdatedTaxProcessingRequest @return @throws FatalException	@param sessionContext @param interestAdjustmentRequestDTO @return @throws FatalException
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	taxAtInterestAdjustment	This method is used to do the tax computation while processing interest adjustment. @param interestAdjustmentRequest @return @throws FatalException	This method is called to deduct on the income specified in the input.  During Interest Adjustment when Tax applicable at Accrual and Accrued Interest is increased then in such scenario the tax   needs to be deducted on Adjusted income . In case the Accrued Interest is decreased then tax won't be refunded.  This method will called for the following cases :- <ol> <li>Tax at Accrual and Accrued Interest is increased. <li>Tax at Compounding and Comounded Interest is increased. <li>Tax at Capitalization and Capitalized Interest is increased. </ol> @param arrangementNumber @param partyId @param amount @param productClass @param dayCountConventionType @param daysForTaxCalculation @param eventID @param transactionReferenceNumber @param valueDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	taxAtInterestAdjustment	This method is used to do the tax computation while processing interest adjustment. @param interestAdjustmentRequest @return @throws FatalException	
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	taxAtInterestAdjustment	This method is used to do the tax computation while processing interest adjustment. @param interestAdjustmentRequest @return @throws FatalException	@param sessionContext @param interestAdjustmentRequestDTO @return @throws FatalException
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackdatedDebitTransaction		
ITaxationCoreAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationCoreAdapter	doBackdatedDebitTransaction		@param sessionContext @param bacRequestDTO @return @throws FatalException
IWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\workflow\IWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntityDTO @throws FatalException	This method calls domain level service for creating workItem and based on the response   triggers a call to initiate common approval process. The initiation of common approval  process is the point where a call will be made to the BPEL client to initiate a process   in BPEL. The initiation internally happens through a web service call  Initiating approval process after committing data in flx_wl_workitem and flx_wl_workitem_dtls table.  If commit fails approval process will not be initiated.  If approval process initiation fails, the committed row from both the tables will be deleted. @param sessionContext @param workItemDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param status @param isModifiedByCreator @throws FatalException
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This method will create the workflow tracker summarized entity. @param workItemDTO @param workItemId @param approvalProcessName
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the statuses in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This operation is used for approving a WorkItem. @param sessionContext SessionContext @param requestDTO WorkItemApprovalRequestDTO, this DTO indicates the ID of the WorkItem, comment for approvals and ExecutableWorkflowStage IDs that the user is approving. @return WorkItemApprovalResponse The response contains the status of the WorkItem and the status of the service request after approvals. It also contains IDs of ExecutableWorkflowStage that the needs approvals next. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the statuses in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This operation will be used for rejecting a workItem from the approval process.  WorkItems raised for approval will be sent to certain users which belong to   predefined approval groups. These users can approve or reject a transaction based  on certain predefined criteria/logic which will be user/transaction specific.  Whenever a user rejects a transaction this method will be invoked and the transaction  will be marked as REJECTED BY INITIATOR/REJECTED BY APPROVER based upon who has  rejected the transaction.  @param sessionContext SessionContext @param requestDTO WorkItemApprovalRequestDTO, this DTO indicates the ID of the WorkItem, comment for approvals and ExecutableWorkflowStage IDs that the user is approving. @return WorkItemApprovalResponse The response contains the status of the WorkItem and the status of the service request after approvals. It also contains IDs of ExecutableWorkflowStage that the needs approvals next. @throws FatalException
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the statuses in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the statuses in the workflow tracker syummarized entity. @param workItemId @param oldWorkItemStatus @param newWorkItemStatus @param processStatus
IWorklistWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IWorklistWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the statuses in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	assertLimits		This method returns the implementation of the asserter for providing user limit authorization logic. The implementation internally handles same branch and inter branch authorization checks based on the inputs data provided by the calling program. Initially through AuthorizationAssertionAdapter, it provides asserters for the service.  Then initialization for various attributes is done by namePairValues.  assertAuthorization checks User Limit authorization and accordingly set the result whether  authorization is required or not required.  WorkFlow retrieved is for particular limit range is stored in FCRTThreadAttribute hashMap @fcb.param MI,IServiceSecurityPolicy,policy, , Policy on the Service. @fcb.param MI,IRequiredAuthorization,requiredAuth, , required Authorization on Service. @return
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	assertLimits		@param transactionAmount @param transactionCurrency @param partyId @return @throws FatalException
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getPartyId		This method logs the customer limits. @fcb.param MI,HashMap<String,Object>,serviceInputs, , Indicates the service inputs. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getPartyId		This method returns the implementation of the asserter for providing user limit authorization logic. The implementation internally handles same branch and inter branch authorization checks based on the inputs data provided by the calling program. Initially through AuthorizationAssertionAdapter, it provides asserters for the service.  Then initialization for various attributes is done by namePairValues.  assertAuthorization checks User Limit authorization and accordingly set the result whether  authorization is required or not required.  WorkFlow retrieved is for particular limit range is stored in FCRTThreadAttribute hashMap @fcb.param MI,IServiceSecurityPolicy,policy, , Policy on the Service. @fcb.param MI,IRequiredAuthorization,requiredAuth, , required Authorization on Service. @return
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getPartyId		This method retrieves the owner partyId for the given account number. @throws FatalException @return,String, Returns the party Id.
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getOwnerPartyId		This method logs the customer limits. @fcb.param MI,HashMap<String,Object>,serviceInputs, , Indicates the service inputs. @throws FatalException
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getAccountId		This method logs the customer limits. @fcb.param MI,HashMap<String,Object>,serviceInputs, , Indicates the service inputs. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getAccountId		This method returns the implementation of the asserter for providing user limit authorization logic. The implementation internally handles same branch and inter branch authorization checks based on the inputs data provided by the calling program. Initially through AuthorizationAssertionAdapter, it provides asserters for the service.  Then initialization for various attributes is done by namePairValues.  assertAuthorization checks User Limit authorization and accordingly set the result whether  authorization is required or not required.  WorkFlow retrieved is for particular limit range is stored in FCRTThreadAttribute hashMap @fcb.param MI,IServiceSecurityPolicy,policy, , Policy on the Service. @fcb.param MI,IRequiredAuthorization,requiredAuth, , required Authorization on Service. @return
ILimitsAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ILimitsAssertionAdapter	getAccountId		This method retrieves the account Id. @throws FatalException @return,String, Returns the account Id.
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		This method determines isFeeNeogotiated based on priceTicketNumber.  @return isFeeNegotiated
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		@return isFeeNegotiated
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	isBaseAmountNegotiated		
ISecurityPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPricingAdapter	fetchPricingDetails		
ICollectionDateManagerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionDateManagerAdapter	isWorkingDay	@param inputDate @return @throws Exception	This method returns the true if Date is a working date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	<pre> This operation is responsible for opening a Loan Account as well configuration of Preferences and facilities associated with a Loan Account. They are :- (1)Account Preferences (2)Accelerated Payment Instruction (3)Offset Account (4)Account Payment Instruction (5)Repayment Schedule Configuration (6)Rate Lock (7)Loan Account Details These details are captured by the Origination system and the details are passed to Loans module as part of the "Account Creation" process.The method works as follows :- (1)Invoke openLoanAccount method which will create a Loan Account (2)Fetch the AccountPreference details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke LoanAccountPreferenceApplicationService to save the Account Preference (3)Fetch the Accelerated Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAcceleratedPaymentInstructionDTO. Invoke LoanAcceleratedPaymentInstructionApplicationService to save the LoanAcceleratedPaymentInstruction (4)Fetch the OffsetAccount details from LoanAccountConfigurationDTO Set the accountId in the OffsetAccountDTO. Invoke OffsetAccountApplicationService to save the OffsetAccount (5)Fetch the Account Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke Account Payment InstructionApplicationService to save the Account Payment Instruction (6)Fetch the Repayment Schedule Configuration details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanRepaymentScheduleApplicationService to save the Repayment Schedule Configuration &lt;/pre&gt;  @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,LoanAccountConfigurationDTO,loanAccountConfigurationDTO. This holds the details required when opening and configuring a loans account. The transaction status will indicate whether the transaction went through successfully and causes the error if unsuccessful. &lt;b&gt;Success &lt;/li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_SUCCESS}&lt;/li&gt; &lt;b&gt;Transaction failure Error Reason may be one of the following {@link com.ofss.fc.common.FAPIErrorConstants#MID_EBA_PROD_NOTMAPPED} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EGL_TXN_NO_INV}&lt;/li&gt; &lt;/ul&gt; @return {@link com.ofss.fc.app.loan.dto.account.LoanAccountOpeningResponse} This response object holds information of the newly generated account and the status of transaction. @throws FatalException in case of error.Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangement Linkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	This method would add/update the broker/introducer arrangement details,DemandDepositAccountOpeningDTO would contain the details of agent arrangement details which is captured by origination during account opening. @param loanAccountConfigurationDTO @param loanAccountOpeningResponse @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	{@link AgentArrangementLinkageApplicationService#addAgentArrangementLinkage(com.ofss.fc.app.context.SessionContext, AgentArrangementLinkageDTO)}
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	This does the processing of term & amount & stage frequency changes.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	<pre> This method provides external modules to open Demand Deposit Accounts. The CASA account opening service will be called during Create Account stage of Origination Following conditions should be checked while CASA account opening. Party Role/(Valid-Merge)Status/Restrictions Minor Major check Party KYC Status Documentation incomplete Permissible number of accounts for a Party under the product Product code (active/expired) Party type Branch Product Xref ALBPC min-max validations Permitted Relationships Casa account opening service will be responsible for Generating and creating account with zero balances If configuration of account number generation allows then account number provided in input will be used for account opening initialize minimum average balance Setting account status based on KYC and document completion status creating pricing plan registering interest plan and logging settlement date change action registering the account with the tax module. The following is the algorithm for the method: Convert the information provided by the calling method via DemandDepositAccountOpeningDTO to domain object using </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DDAccountOpeningAssembler#toDomainObject()&amp;lt;/li&gt; &amp;lt;b&gt;Changes since: R2.2</p> <p>Check the account opening date in the DemandDepositAccountOpeningDTO is before the current posting date for the current branch and bank,if it is true then call</p> <p>Call the domain service</p> <p>DDAccountOpeningService#openDemandDepositAccountBackdated(DemandDepositAccount, String, PartyToAccountRelationship[], boolean, boolean()) to validate and create the account</p> <p>DDAccountOpeningService#openDemandDepositAccountBackdated(DemandDepositAccount, String, PartyToAccountRelationship[], boolean, boolean()) internally will also create pricing plan backdated,register interest plan backdated,intimate tax backdated. else</p> <p>Call the domain service</p> <p>DDAccountOpeningService#openDemandDepositAccount() to validate and create the account</p> <p>DDAccountOpeningService#openDemandDepositAccount() internally will also create pricing plan and register interest plan.</p> <p>Fetch the agent(broker/introducer) arrangement details who has helped in opening the account,these details are passed via</p> <p>DemandDepositAccountOpeningDTO} and stamp it for the given account.</p> <p>Call the alert module by</p> <p>this#callAlert() to log message that the account is opened.</p> <p>The following are the parameters taken by the method along with their significance</p> <p></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demandDepositAccountOpeningDTO : The DTO containing information for account opening, passed by the calling module [eg: Origination], to this method.</p> <pre> {@link DemandDepositAccountOpeningDTO}&amp;lt;/li&gt;&amp;gt; &amp;lt;/li&gt;&amp;gt;@return DDAccountOpeningResponse : The response object which is returned after account opening. See {@link DDAccountOpeningResponse} for more info&amp;lt;/li&gt;&amp;gt; &amp;lt;/ul&gt;&amp;gt; See {@link DDAccountOpeningService#openDemandDepositAccount(domainObject: IDemandDepositAccount) } and {@link DemandDepositAccount} for more information. &lt;/pre&gt; @param demandDepositAccountOpeningDTO @throws FatalException </pre>
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	This method would add/update the broker/introducer arrangement details,DemandDepositAccountOpeningDTO would contain the details of agent arrangement details which is captured by origination during account opening. @param sessionContext @param demandDepositAccountOpeningDTO @param transactionStatus @param response @return @throws FatalException
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAgentAdapter	addAgentArrangementLinkage	This method is called after successful account opening from origination to register the agent commission details. {@link AgentArrangementLinkageApplicationService#} @param serialNo @param accountId @return taskModuleType	This method would add/update the broker/introducer arrangement details,DemandDepositAccountOpeningDTO would contain the details of agent arrangement details which is captured by origination during account opening. @param sessionContext @param demandDepositAccountOpeningDTO @param transactionStatus @param response @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	fetchLatestArrangementForAgent	This method returns agent details for an accountId or arrangementId @param sessionContext @param arrangementId @return @throws FatalException	This method invokes agentArrangement service to fetch agent details
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	fetchLatestArrangementForAgent	This method returns agent details for an accountId or arrangementId @param sessionContext @param arrangementId @return @throws FatalException	
IAccountAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAgentAdapter	fetchLatestArrangementForAgent	This method returns agent details for an accountId or arrangementId @param sessionContext @param arrangementId @return @throws FatalException	@param sessionContext @param accountId @return @throws FatalException
IDDASMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASMSAdapter	fetchAllNonFinancialTaskCodes	Method to return non-financial task codes @param sessionContext @return List<String> @throws FatalException	
IDDASMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASMSAdapter	fetchAllNonFinancialTaskCodes	Method to return non-financial task codes @param sessionContext @return List<String> @throws FatalException	
IDDASMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASMSAdapter	checkPartyLevelLimitsInBatch	This method checks the party level limits defined in SMS under a group.  If the limit is expired system will throw exception, which will dishonour the transaction.  @param transactionAmount @param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This method will create the Workflow Tracker Summarized Entity. @see com.ofss.fc.app.adapter.origination.IOriginationWorkflowAdapter#createWorkflowTrackerSummarizedEntity(com.ofss.fc.app.workflow.tracker.WorkflowTrackerSummarizedEntityDTO)
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This service will create a submission. @fcb.param MI,ApplicationContext,applicationContext, , ApplicationContext which must contain the application context information such as transaction branch details, process date, etc. @fcb.param MI, Submission, submission, , Indicates a group of applications applied by a set of applicants. Each application contains various product groups for which accounts are opened. @return Submission, Returns the entity that is created
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO @throws FatalException	This method will create the workflow tracker summarized entity. @param submission
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the Workflow Tracker Summarized Entity. @see com.ofss.fc.app.adapter.origination.IOriginationWorkflowAdapter#updateWorkflowTrackerSummarizedEntity(com.ofss.fc.app.workflow.tracker.WorkflowTrackerSummarizedEntityDTO)
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the status in the workflow tracker summarized entity. @param submission @param submissionStatusType
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the submissionStatus for Batch processing using HQL @fcb.param MI, String, submissionId, , Indicates the identifier for the submission @fcb.param SubmissionStatusType, submissionStatus, , Indicates the submission status to be updated. @fcb.param Date, completionDate, , Indicates the completion date, will be present only when submission status is SubmissionStatusType.SUBMISSION_COMPLETED
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the existing submission summarized entity. @param submissionSummarizedEntity @throws PersistenceException
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the workflow tracker summarized entity. @param submissionId @param submissionStatusType
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the status in the workflow tracker summarized entity. @param submission

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the status in the workflow tracker summarized entity. @param submission @param submissionStatusType
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	@param submissionSummarizedEntity
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	processStageChangeEvent	This method is used to process a application stage change event @param sessionContext @param stageChangeEventDTO @return {@link TransactionStatus} @throws FatalException	This method is used to process a application stage change event @param sessionContext @param stageChangeEventDTO @return {@link TransactionStatus} @throws FatalException
IOrigination WorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationWorkflowAdapter	processStageChangeEvent	This method is used to process a application stage change event @param sessionContext @param stageChangeEventDTO @return {@link TransactionStatus} @throws FatalException	This method will invoke the app service (via workflow adapter) which handles the processing of an application stage change event. @param sessionContext the {@link SessionContext} @param processKey the key identifier @param stageId the Stage Identifier @param start indicates whether its the start or end of stage @throws FatalException
ICalendarServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\calendar\ICalendarServiceAdapter	isValidPostingDate	This method returns whether the given posting date is a valid working date for the given branch. @param branchCode - a valid branch code or composite branch (0) @param postingDate - the current working transaction date. @return boolean- returns true if it is a working day. @throws FatalException {@link	This method returns the true if Date is a working date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			com.ofss.fc.infra.exception.FatalException}	
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	fetchInterestPlanItemDetails		
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	fetchInterestPlanItemDetails		This method will return all the details of a deposit that need to be displayed in the statement inquiry screen. @param sessionContext @param accountId @param depositNumber @param interestClass @param interestDivision @param effectiveDate @throws FatalException
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	writeOffAccount	This method calls the This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId	This method calls the This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	fetchRevisedActionDates	The method will fetch recalculated action due date along with its holiday treatment. This will be called from {@link ActionSetRearranger} when interest actions are relogged on calendar change. @param arrangementNumber @param actionType @return	The method will re-log the actions present in {@link ActionSetRearrangerBatchData#getActionSequence()} based on its module type. The method will call the corresponding {@link ActionLogger} methods to log the action again. <p> For interest related actions i-e CreditInterestCapitalization,CreditInterestSettlement, DebitInterestCapitalization,DebitInterestSettlement the holiday treatment and adjusted dueDates will be fetched from Interest module. Also BalanceChange action will be logged with Current process date as the dueDate so as to avoid trying to achieve balance change on a holiday. </p> @param applicationContext @param actionSetBatchData



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountInterestAdapter	fetchRevisedActionDates	The method will fetch recalculated action due date along with its holiday treatment. This will be called from { <a href="#">@link ActionSetRearranger</a> } when interest actions are relogged on calendar change. @param arrangementNumber @param actionType @return	Fetch interest details. @param actionSetBatchData @param processDate @param interestClass @param interestDivision @param effectiveDate @param interestPhase @return
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountInterestAdapter	calculateInterestBenefit	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountInterestAdapter	calculateInterestBenefit	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	checkIfInterestAdjustmentsDoneForTheDate	This method is used to check if the interest adjustment is done for the day for particular account.   Since R2.2</br> @param accountId @param moduleType @param depositNumber @param transferDate @return	This method is used to check if the interest adjustment is done for the day for particular account.   Since R2.2</br> @param accountId @param depositNumber @param transferDate @return
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	checkIfInterestAdjustmentsDoneForTheDate	This method is used to check if the interest adjustment is done for the day for particular account.   Since R2.2</br> @param accountId @param moduleType @param depositNumber @param transferDate @return	Method validateInterestAdjustmentOnAccount is called to check whether there is any interest adjustment occurred on the account for the given posting date.   Since R2.2</br> @param moduleType
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	inquireInterestBenefit	fetch interest benefit. @param arrangementNumber @param currency @param startDate @param endDate @return	
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	inquireInterestBenefit	fetch interest benefit. @param arrangementNumber @param currency @param startDate @param endDate @return	This method calls the interest service and calculates the cumulative offset balance till the last charging date. @param sessionContext @param arrangementNumber @param currency @param startDate @param endDate @return @throws FatalException
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	inquireInterestBenefit	fetch interest benefit. @param arrangementNumber @param currency @param startDate @param endDate @return	Method will be used to enquire total offset benefit enjoyed by account due to offset facility. @param benefitInquiryRequest @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountInterestAdapter	initiateBackdatedOffsetBenefitCalculation	This method will initiate the backdated offset benefit calculation. This method will <ol style="list-style-type: none"> <li>Move existing offset benefit log items to history</li> <li>Fetch list of periods for which the benefit amount has to be recalculated.</li> </ol> @param accountNumber @param productClass @param backdatedValueDate @param isNewCharging @return	
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountInterestAdapter	initiateBackdatedOffsetBenefitCalculation	This method will initiate the backdated offset benefit calculation. This method will <ol style="list-style-type: none"> <li>Move existing offset benefit log items to history</li> <li>Fetch list of periods for which the benefit amount has to be recalculated.</li> </ol> @param accountNumber @param productClass @param backdatedValueDate @param isNewCharging @return	This method will be executed in batch. It will compute the benefit occurred on account and will dump it. The steps are as follows: <ol style="list-style-type: none"> <li>If <a href="#">AccountBenefitActionType#Offset</a> then, create the <a href="#">InterestBenefitRequest</a> by calling <a href="#">DDAOffsetMonetaryBenefit#createBenefitRequest</a> or <a href="#">LoanOffsetMonetaryBenefit#createBenefitRequest</a> and call the <a href="#">OffsetBenefitAnalyzer#calculateBenefit(InterestBenefitRequest)</a> which eventually calls <a href="#">InterestBenefitService</a>. <li>Else if <a href="#">AccountBenefitActionType#BackdatedOffset</a>, then <ul style="list-style-type: none"> <li>if <code>isNewCharging</code> is true then call the <a href="#">AbstractBenefitAnalyzer#reverseInterestBenefitAmounts(String, Date)</a> to reverse the interest benefit amounts. <li>Get the list of <a href="#">Period</a> by calling <a href="#">AccountInterestAdapter#initiateBackdatedOffsetBenefitCalculation</a> which finally calls the <a href="#">InterestBenefitService</a>. <li>Iterate over the <a href="#">Period</a> list and <a href="#">createBenefitRequest</a> and send this request to the <a href="#">AbstractBenefitAnalyzer#calculateBenefit(BenefitRequest)</a></li> </li></li></ul> </li> </li></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				est)) to obtain the response. @param accountId @param fromDate @param toDate @param isNewCharging @param productClass @return response
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	reverseOffsetBenefitAmounts		
IAccountInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountInterestAdapter	reverseOffsetBenefitAmounts		This method will reverse the offset benefit amounts having end date greater than input from date. This method eventually calls the InterestBenefitService to reverse the Offset Benefit Amounts.If isNewCharging is true which is in some cases of Loans, thenThis method is called. This method internally calls InterestBenefitService which deletes the interestBenefitLogItems present from the backdatedValueDate provided. The list of {@link InterestBenefitResponse} is sent from the InterestBenefitService which is iterated over and used for logging Non Financial Statement Item. @param accountId @param fromDate
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\batch\IBatchAdapter	updateFinYearOrPeriodCodeForValidBranches		This method will update the financial year and the period code in the flx_cs_branches_b table after the EOD category has been completed. This method will call {@link CoreService#updateFinYearOrPeriodCodeForBranch(String, String, Date)} to update the fields. @param branchGroupCode @param batchCategory @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\batch\IBatchAdapter	updateFinYearOrPeriodCodeForValidBranches		<p>Method callApEodEpilogue. Called at the end of each shell or category if there are no further shells to start. Algorithm</p> <ol style="list-style-type: none"> <li>If category is <a href="#">{@link ProcessCategoryType#END_OF_DAY}</a>, validate whether cutoff has run or not. If not run return failure.</li> <li>Fetch number of shells in completed,aborted, running, not started state by calling <a href="#">{@link #fetchShellStatusCount(String, String)}</a>.</li> <li>If number of shells in running state &gt; 0 , return success as the category is yet to be completed</li> <li>If number of aborted shells &gt; 0, update <a href="#">{@link BatchCategory}</a> with status as Aborted else update <a href="#">{@link BatchCategory}</a> with status as completed</li> <li>If all the shells are completed <ol style="list-style-type: none"> <li>Delete <a href="#">{@link GroupBranchXref}</a> by calling <a href="#">{@link #deleteGroupBranchXref(String, String)}</a> method.</li> <li>Update the shell status by calling <a href="#">{@link #updateShellsOnCategoryCompletion(String, String)}</a> method.</li> <li>Update the previous, current and next run dates in <a href="#">{@link BranchGroupCategory}</a> by calling <a href="#">{@link #updateBranchGroupCategory(BranchGroupCategory, int)}</a></li> <li>If category is <a href="#">{@link ProcessCategoryType#END_OF_DAY}</a>, <ol style="list-style-type: none"> <li>Update the financial year codes for branches by calling <a href="#">{@link #updateFinYearOrPeriodCodeForValidBranches(String, String)}</a></li> <li>Update the branch dates by calling method <a href="#">{@link #updateBranchDates(String, String)}</a>.</li> <li>If all branch groups have completed EOD, update the bank dates by calling <a href="#">{@link #checkAndUpdateBankDates(BranchGroupCategory)}</a>.</li> </ol> </li> </ol> </li> </ol> <p>@param p_xml_bp</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchAdapter	updateBankDates	This method will update the dates in the {@link BankDates} entity specified by the input bank code. @param bankCode @param previousWorkingDate @param currentWorkingDate @param nextWorkingDate	
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchAdapter	updateBankDates	This method will update the dates in the {@link BankDates} entity specified by the input bank code. @param bankCode @param previousWorkingDate @param currentWorkingDate @param nextWorkingDate	This method will update the bank dates if applicable. This method will first check whether all the branch groups in the bank have completed EOD or not. If EOD has been completed for branch groups, then this method will update the previous , current and next working dates for the bank in the (flx_cs_bank_dates) table. @param branchGroupCategory
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchAdapter	publishAbortMessage		This method is used to publish Success mail for Category to the specified caller in Batch.properties.  In case the Category get's completed Alert can be send to the Specified Caller configured in Batch.properties.  @param p_xml_bp @param shellname
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchAdapter	publishAbortMessage		This method will mark the shell and category as aborted. This method will not create the {@link ShellExecutionHistory} and {@link BatchCategoryExecutionHistory} logs. @param p_xml_bp @param abortCategory @throws Exception
IBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchAdapter	publishAbortMessage		This method is used to publish Batch abort mail to the specified caller in Batch.properties.  In case the Batch get aborted whether to send Alert or not is configured in Batch.properties.  Once the Batch alert is configured mails will be send to the Callers specified in the Batch.properties.  @param p_xml_bp @param shellname

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IProductAssetClassificationAdapter	applyPlanIdRestrictions	@return @throws com.ofss.fc.infra.exception.FatalException	@param loanProduct
IProductAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IProductAssetClassificationAdapter	applyPlanIdRestrictions	@return @throws com.ofss.fc.infra.exception.FatalException	<p>This method will check the asset classification parameter for the input data if debit balance allowed is checked. </p> @param ddaProduct
IProductAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IProductAssetClassificationAdapter	applyPlanIdRestrictionsForLoan	@return @throws com.ofss.fc.infra.exception.FatalException	@param loanProduct

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	retrievePreApprovedEligibility	This method retrieves pre-approved eligibility for a set of applicants and a particular lending or credit card product group. It invokes an external system. The logic for processing pre-approved eligibility varies across external systems. @param applicantDataDTOs @param submissionId @param productGroupLinkageDTO @return PreApprovedLimitDTO	This method processes pre-approved eligibility for a particular lending or credit card product group that is associated with a set of applicants. The main logic of this method is as follows: <ol style="list-style-type: none"> <li>If product group is pre-approved, call the external system to release the pre-approved limit associated with a unique id</li> <li>Call the external system to compute pre approved eligibility for the applicants' information and product group's information passed</li> <li>If product group is pre-approved, call the external system to put hold on the pre-approved limit</li> </ol> @param SessionContext sessionContext, Contains the session context information such as user id, branch code, channel, etc. @param SubmissionInputDTO submissionInputDTO, Contains the submission details such as applicant and facility information. @param Integer pgSerialNumber, Identifies a particular ProductGroupLinkageDTO along with applicationId . @return {@link com.ofss.fc.app.origination.dto.lending.core.preapprovedlimit.PreApprovedLimitDTO}, Contains pre approved limit reference id and pre approved status @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	retrievePreApprovedEligibility	This method retrieves pre-approved eligibility for a set of applicants and a particular lending or credit card product group. It invokes an external system. The logic for processing pre-approved eligibility varies across external systems. @param applicantDataDTOs @param submissionId @param productGroupLinkageDTO @return PreApprovedLimitDTO	This method invokes the external system to re-assess the pre-approved eligibility of an existing submission's lending or credit card product group that was pre-approved. It is an attempt to re-hold the pre-approved limit in an existing submission's product group that becomes released during edit submission. The rationale is to not lose the pre-approved limit of an existing submission's product group during edit submission via removing the product group or changing details of the product group. This aids cancel edit during edit submission to return to the initial submission's original states. @param sessionContext @param initialSubmissionProductGroupLinkageDTO @param initialSubmissionProductGroupApplicantDataDTOs
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	updatePreApprovedEligibility	This method updates pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. If isPreApproved flag passed is true, it indicates to the external system that the pre-approved limit should be marked as held or in use. Otherwise, isPreApproved flag passed is false, it indicates to the external system that the pre-approved limit should be marked as not held or not in use. @param preApprovedLimitRefId @param isPreApproved @return	This method processes pre-approved eligibility for a particular lending or credit card product group that is associated with a set of applicants. The main logic of this method is as follows: <ol style="list-style-type: none"><li>If product group is pre-approved, call the external system to release the pre-approved limit associated with a unique id</li><li>Call the external system to compute pre approved eligibility for the applicants' information and product group's information passed</li><li>If product group is pre-approved, call the external system to put hold on the pre-approved limit</li></ol> @param SessionContext sessionContext, Contains the session context information such as user id, branch code, channel, etc. @param SubmissionInputDTO submissionInputDTO, Contains the submission details such as applicant and facility information. @param Integer pgSerialNumber, Identifies a particular ProductGroupLinkageDTO along with applicationId . @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.origination.dto.lending.core.preapprovedlimit.PreApprovedLimitDTO}, Contains pre approved limit reference id and pre approved status @throws FatalException
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	updatePreApprovedEligibility	This method updates pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. If isPreApproved flag passed is true, it indicates to the external system that the pre-approved limit should be marked as held or in use. Otherwise, isPreApproved flag passed is false, it indicates to the external system that the pre-approved limit should be marked as not held or not in use. @param preApprovedLimitRefId @param isPreApproved @return	This method calls the external system to mark a particular pre-approved limit that is associated with a product group and set of applicants as in use or held. @param sessionContext @param productGroupLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	updatePreApprovedEligibility	This method updates pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. If isPreApproved flag passed is true, it indicates to the external system that the pre-approved limit should be marked as held or in use. Otherwise, isPreApproved flag passed is false, it indicates to the external system that the pre-approved limit should be marked as not held or not in use. @param preApprovedLimitRefId @param isPreApproved @return	This method calls the external system to release hold for a pre-approved limit that is associated with a product group and set of applicants. @param sessionContext @param submissionId @param productGroupLinkageDTO @return @throws FatalException
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	updatePreApprovedEligibility	This method updates pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. If isPreApproved flag passed is true, it indicates to the external system that the pre-approved limit should be marked as held or in use. Otherwise, isPreApproved flag passed is false, it indicates to the external system that the pre-approved limit should be marked as not held or not in use. @param preApprovedLimitRefId @param isPreApproved @return	@see IOriginationPreApprovedLimitAdapter#updatePreApprovedEligibility(String, Boolean)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	confirmPreApprovedEligibility	This method confirms pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. It indicates to the external system that the pre-approved limit should be marked as utilized. Subsequently, if an identical set of applicants along with a product group enquires the external system on pre-approved eligibility, the external system would return false since the limit has been utilized. Even if the product group's requested amount is below the external system's allocated pre-approved limit, the entire pre-approved limit is marked as utilized. @param applicantDataDTOs @param productGroupLinkageDTO @return PreApprovedLimitDTO	This service calls the external system to mark a particular pre-approved limit that is associated with a product group and set of applicants as utilized. This method gets called upon: <ul> <li>Lending or credit card account opening. </ul> @fcb.param MI, SessionContext sessionContext, Contains the session context information such as user id, branch code, channel, etc. @fcb.param MI, String preApprovedLimitRefId, Identifier from the external system on pre-approved eligibility for product group and set of applicants. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPreApprovedLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPreApprovedLimitAdapter	confirmPreApprovedEligibility	This method confirms pre-approved eligibility for a particular pre approved limit reference id that is associated with a set of applicants and a lending or credit card product group. It invokes an external system. It indicates to the external system that the pre-approved limit should be marked as utilized. Subsequently, if an identical set of applicants along with a product group enquires the external system on pre-approved eligibility, the external system would return false since the limit has been utilized. Even if the product group's requested amount is below the external system's allocated pre-approved limit, the entire pre-approved limit is marked as utilized. @param applicantDataDTOs @param productGroupLinkageDTO @return PreApprovedLimitDTO	@see IOriginationPreApprovedLimitAdapter#confirmPreApprovedEligibility(SessionContext, String)
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearchagent\ITitleSearchAgentEligibilityAdapter	addRestrictionLinkageAndException		This method adds a new Restriction Linkage And Exception. @return transactionStatus @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ITitleSearchAgentEligibilityAdapter#addRestrictionLinkageAndException(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.commonservice.core.dto.restriction.RestrictionExceptionAdditionDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearchagent\ITitleSearchAgentEligibilityAdapter	fetchAllAllowableValues		Returns the <b>Eligible</b> titleSearchAgents This will call the Eligibility Framework with the collateral Facts, and the call will return the list of TitleSearchAgents that are applicable for the TitleSearch. @fcb.param SessionContext,sessionContext @fcb.param String,collateralID @return {@link com.ofss.fc.app.lcm.dto.collaterals.titlesearch.FetchAgentInquiryResponse} @throws FatalException
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearchagent\ITitleSearchAgentEligibilityAdapter	fetchAllAllowableValues		This method fetches all allowable values . @return AllAllowableValues @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ITitleSearchAgentEligibilityAdapter#fetchAllAllowableValues(com.ofss.fc.app.context.SessionContext, com.ofss.fc.enumeration.commonservice.DomainEntityType, com.ofss.fc.app.commonservice.core.dto.restriction.PreliminaryRestrictionDTO[])
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	addRestrictionLinkageAndException		This method <u>Adds</u> a new Valuer Preference. @fcb.param SessionContext,sessionContext @fcb.param CollateralValuerPreferenceDTO,collateralValuerPreferenceDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException @see com.ofss.fc.app.lcm.service.collaterals.valuation.valuer.preference.ICollateralValuerPreferenceApplicationService#addCollateralValuerPreference(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.valuation.valuer.preference.CollateralValuerPreferenceDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	addRestrictionLinkageAndException		This method <u>updates</u> the status of the existing Valuer preference. @fcb.param SessionContext,sessionContext @fcb.param CollateralValuerPreferenceDTO,collateralValuerPreferenceDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException @see com.ofss.fc.app.lcm.service.collaterals.valuation.valuer.preference.ICollateralValuerPreferenceApplicationService#updateCollateralValuerPreference(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.valuation.valuer.preference.CollateralValuerPreferenceDTO)
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	addRestrictionLinkageAndException		
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	addRestrictionLinkageAndException		This method <u>Adds</u> a new Agent for the Colateral TitleSearch. @fcb.param SessionContext, sessionContext @fcb.param CollateralTitleSearchAgentPreferenceDTO, collateralTitleSearchAgentPreferenceDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException @see com.ofss.fc.app.lcm.service.collaterals.titlesearch.agent.preference.ICollateralTitleSearchAgentPreferenceApplicationService#addCollateralTitleSearchAgentPreference(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.titlesearch.agent.pr

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ference.CollateralTitleSearchAgentPreferenceDTO)
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	addRestrictionLinkageAndException		This method <u>updates</u> the value of existing AgentID for Collateral TitleSearch. @fcb.param SessionContext,sessionContext @fcb.param CollateralTitleSearchAgentPreferenceDTO, collateralTitleSearchAgentPreferenceDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException @see com.ofss.fc.app.lcm.service.collaterals.titlesearch.agent.preference.ICollateralTitleSearchAgentPreferenceApplicationService#updateCollateralTitleSearchAgentPreference(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.titlesearch.agent.preference.CollateralTitleSearchAgentPreferenceDTO)
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAdapter	fetchAllAllowableValues		This method is used to fetch the <b>Eligible valuers</b> This will call the Eligibility Framework with the collateral Facts , and the call will return the <b>list of Valuers</b> that are applicable for the valuation. @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationDTO collateralValuationDTO : Collateral valuation request details DTO. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.valuer.fetchvaluer.FetchValuerInquiryResponse} This class holds the values of all the DTOs defined in it.  @throws FatalException
IValuerEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\IValuerEligibilityAd	fetchAllAllowableValues		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	apter			
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		<p>This method is used to reverse withdraw cash from a given original transaction reference number identified by originalTransactionReferenceNumber. The withdrawal count is decreased by one as the transaction is reversed. &lt;ol&gt; The reverse cash withdrawal method works as follows :&lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;validate if transaction is allowed to reversed by calling  super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String), Pass the original transaction reference number from session context.&lt;/li&gt;</li> <li>&lt;li&gt;call method  super.reverseDebitFunds(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String) to reverse debited funds. &lt;/li&gt;</li> <li>&lt;li&gt;call super.manageReverseFees(sessionContext: SessionContext) for reversing the fee.&lt;/li&gt;</li> <li>&lt;li&gt;Log reverse financial accounting event using {&lt;link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext)&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;  @param sessionContext @param originalTransactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		<p>This method is used to reverse deposit cash from a given original transaction reference number identified by <code>sessionContext.getOriginalTransactionReferenceNumber</code>. <ol style="list-style-type: none"> <li>The reverse cash deposit method works as follows : <ul style="list-style-type: none"> <li>validate if transaction is allowed to be reversed by calling <code>super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String)</code>, Pass the original transaction reference number from session context where the <code>eventId</code> and <code>mnemonicCode</code> are constants.</li> <li>Fetch transaction audit items using original transaction reference number by calling <code>{@link TransactionAuditItemRepository#fetchTransactionAuditItems(transactionReferenceNumber: String, eventId:String, mnemonicCode:String):TransactionAuditItem[1..]}</code> <ul style="list-style-type: none"> <li>call private method <code>reverseCreditFunds(transactionAuditItems:TransactionAuditItem[])</code> to reverse debited funds.</li> <li>call <code>super.manageReverseFees(sessionContext:SessionContext)</code> for reversing the fee.</li> <li>Log reverse financial accounting event using <code>{@link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext)}</code></li> </ul> </li> </ul> </li> </ol> <p>@param sessionContext @throws FatalException</p> </p>
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		<p>Method will be used to notify BROP for cash transaction reversal so that cash position can be adjusted accordingly. @param sessionContext @param originalReferenceNumber @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		<p>This method is used to reverse deposit cash from a given original transaction reference number identified by <code>sessionContext.getOriginalTransactionReferenceNumber</code>. <ol style="list-style-type: none"> <li>The reverse cash deposit method works as follows : <ul style="list-style-type: none"> <li>validate if transaction is allowed to reversed by calling <code>super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String)</code>, Pass the original transaction reference number from session context wheres the <code>eventId</code> and <code>mnemonicCode</code> are constants.</li> <li>Fetch transaction audit items using original transaction reference number by by calling <code>{@link TransactionAuditItemRepository#fetchTransactionAuditItems(transactionReferenceNumber: String, eventId:String, mnemonicCode:String):TransactionAuditItem[1..]}</code> <ul style="list-style-type: none"> <li>call private method <code>reverseCreditFunds(transactionAuditItems: TransactionAuditItem[])</code> to reverse debited funds.</li> <li>call <code>super.manageReverseFees(sessionContext: SessionContext)</code> for reversing the fee.</li> <li>Log reverse financial accounting event using <code>{@link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext)}</code></li> </ul> </li> </ul> </li> </ol> <p>@param sessionContext @return @throws FatalException</p> </p>
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPosition		<p>This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPositionForReversal	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException	This method is used to reverse withdraw cash from a given original transaction reference number identified by originalTransactionReferenceNumber. The withdrawal count is decreased by one as the transaction is reversed. <ol> The reverse cash withdrawal method works as follows :  <li>validate if transaction is allowed to reversed by calling super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String), Pass the original transaction reference number from session context.</li> <li>call method super.reverseDebitFunds(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String) to reverse debited funds. <li>call super.manageReverseFees(sessionContext: SessionContext) for reversing the fee.</li> <li>Log reverse financial accounting event using (@link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext))</li> </ol> @param sessionContext @param originalTransactionReferenceNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPositionForReversal	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException	This method is used to reverse deposit cash from a given original transaction reference number identified by sessionContext.getOriginalTransactionReferenceNumber. <ol> The reverse cash deposit method works as follows :  <li>validate if transaction is allowed to reversed by calling super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String), Pass the original transaction reference number from session context wheres the eventId and mnemonicCode are constants.</li> <li>Fetch transaction audit items using original transaction reference number by by calling {@link TransactionAuditItemRepository#fetchTransactionAuditItems(transactionReferenceNumber: String, eventId:String, mnemonicCode:String):TransactionAuditItem[1..]} <li>call private method reverseCreditFunds(transactionAuditItems: TransactionAuditItem[]) to reverse debited funds. <li> <li>call super.manageReverseFees(sessionContext: SessionContext) for reversing the fee.</li> <li>Log reverse financial accounting event using {@link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext)}</li> </ol> @param sessionContext @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPositionForReversal	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException	Method will be used to notify BROP for cash transaction reversal so that cash position can be adjusted accordingly. @param sessionContext @param originalReferenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPositionForReversal	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException	This method is used to reverse deposit cash from a given original transaction reference number identified by sessionContext.getOriginalTransactionReferenceNumber. <ol> The reverse cash deposit method works as follows :  <li>validate if transaction is allowed to reversed by calling super.validateForReversal(originalTransactionReferenceNumber:String, eventId:String, mnemonicCode:String), Pass the original transaction reference number from session context wheres the eventId and mnemonicCode are constants.</li> <li>Fetch transaction audit items using original transaction reference number by by calling {@link TransactionAuditItemRepository#fetchTransactionAuditItems(transactionReferenceNumber: String, eventId:String, mnemonicCode:String):TransactionAuditItem[1..]} <li>call private method reverseCreditFunds(transactionAuditItems: TransactionAuditItem[]) to reverse debited funds. <li> <li>call super.manageReverseFees(sessionContext: SessionContext) for reversing the fee.</li> <li>Log reverse financial accounting event using {@link DomainAccountingAdapter#raiseReverseFinancialEvents(sessionContext: SessionContext)}</li> </ol> @param sessionContext @return @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	notifyCashPositionForReversal	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException	This method will be used to notify BROP to do the cash position adjustment in case of cash transaction reversal. @param sessionContext @param originalReferenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to create CashTransactionDTO that will be used as a input for DemandDepositCashTransactionService </p> @param sessionContext @param fundsTransferRequestDTO @return CashTransactionDTO @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to create CashTransactionDTO that will be used as a input for DemandDepositCashTransactionService </p> @param sessionContext @param fundsTransferRequestDTO @return CashTransactionDTO @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO. </p> @param transactionContext
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method redeems a deposit in a Term Deposit account, the deposit can be redeemed to 1) CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO for the logged in User. </p> @param transactionContext @return CashPositionDTO @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method creates a new deposit in a Term Deposit account funded 1) By CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input. @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO for the logged in User. </p> @param transactionContext @return CashPositionDTO @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method redeems a deposit in a Term Deposit account, the deposit can be redeemed to 1) CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	This method creates a new deposit in a Term Deposit account funded 1) By CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input. @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACashAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACashAdapter	fetchCashPosition	This method would fetch the cash position by calling the BROP service	<p>This is the method that will be used to fetch the CashPositionDTO for the logged in User. </p> @param transactionContext @return CashPositionDTO @throws FatalException
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	searchApplication	This method is used to fetch status of customer's loan application @param sessionContext @param applicationSearchRequestDTO @return @throws FatalException	This method lists notifications for a party. The notifications include: (1) Individual warning (2) Trust warning (3) Corporate warning (4) KYC Pending (5) Photo not received (6) Identity (Driving License/ Passport/ PAN Card) not received (7) Customer loan rejected (8) Customer offer dispatched (9) Customer loan pending @param sessionContext @param notificationMessageTemplateService @param partyNotificationResponse @return @throws FatalException
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	searchApplication	This method is used to fetch status of customer's loan application @param sessionContext @param applicationSearchRequestDTO @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	inquirePreferences	This method is used to fetch minor account status @param sessionContext @param accountId @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchAccountAssetClassification	This method is used to fetch reason of non performance of loan account @param sessionContext @param accountId @return @throws FatalException	This method lists notifications for a loan account. The notifications include: (1) Arrears due (2) Installment due (3) Advance payment (4) Non performing (5) Closure request (6) Account delinquent (7) Term credit balance in closed account @param sessionContext @param notificationMessageTemplateService @param loanAccountNotificationResponses @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchAccountAssetClassification	This method is used to fetch reason of non performance of loan account @param sessionContext @param accountId @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchDelinquencyDetails	This method is used to determine if a loan account is delinquent or not @param applicationContext @param accountId @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	inquireLoanBalance	This method is used to fetch account status @param applicationContext @param accountId @return @throws FatalException	This method lists notifications for a loan account. The notifications include: (1) Arrears due (2) Installment due (3) Advance payment (4) Non performing (5) Closure request (6) Account delinquent (7) Term credit balance in closed account @param sessionContext @param notificationMessageTemplateService @param loanAccountNotificationResponses @return @throws FatalException
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	inquireLoanBalance	This method is used to fetch account status @param applicationContext @param accountId @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchDeposit	This method is used to fetch maturity amount and date for a TD account @param sessionContext @param accountId @param depositId @return @throws FatalException	This method lists notifications for a TD account. The notifications include: (1) Instructions maturing (2) Outflow of funds maturing (3) Maturity Instructions Pending (4) Deposit unclaimed @param sessionContext @param notificationMessageTemplateService @param termDepositNotificationResponses @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchDeposit	This method is used to fetch maturity amount and date for a TD account @param sessionContext @param accountId @param depositId @return @throws FatalException	
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchNotificationDetailsForAccountId	This method is used to check if deposit for a TD has matured and if deposit is going to get moved to unclaimed soon @param sessionContext @param accountId @return @throws FatalException	This method lists notifications for a TD account. The notifications include: (1) Instructions maturing (2) Outflow of funds maturing (3) Maturity Instructions Pending (4) Deposit unclaimed @param sessionContext @param notificationMessageTemplateService @param termDepositNotificationResponses @return @throws FatalException
ILogNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\notification\ILogNotificationAdapter	fetchNotificationDetailsForAccountId	This method is used to check if deposit for a TD has matured and if deposit is going to get moved to unclaimed soon @param sessionContext @param accountId @return @throws FatalException	
IAnalyticsWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\workflow\IAnalyticsWebServicesAdapter	insertRowIntoBAMDataObject	This method will insert a row into the specified BAM Data Object. @param dataObject the fully qualified id of the BAM Data Object @param xmlPayload the attribute values data for the row in the form of XML payload @return a boolean indicating the success of the operation @throws FatalException	This method publishes the given {@link BAMDataObject} to BAM server. It extracts Data Object Path, Comma Separated Key Fields as String and XML Payload from the given data object and invokes the BAM Data object operations web service through the corresponding adapter. @param bamDataObject the {@link BAMDataObject} to be published @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAalyticsWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\workflow\IAalyticsWebServicesAdapter	upsertRowIntoBAMDataObject	This method will upsert (update if present or insert otherwise) a row into the specified BAM Data Object. @param dataObject the fully qualified id of the BAM Data Object @param keysCSV the comma separated string of the key attributes of the specified BAM Data Object @param xmlPayload the attribute values data for the row in the form of XML payload @return a boolean indicating the success of the operation @throws FatalException	This method publishes the given {@link BAMDataObject} to BAM server. It extracts Data Object Path, Comma Separated Key Fields as String and XML Payload from the given data object and invokes the BAM Data object operations web service through the corresponding adapter. @param bamDataObject the {@link BAMDataObject} to be published @throws FatalException
IAalyticsWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\workflow\IAalyticsWebServicesAdapter	updateRowInBAMDataObject	This method will update a row into the specified BAM Data Object. @param dataObject the fully qualified id of the BAM Data Object @param keysCSV the comma separated string of the key attributes of the specified BAM Data Object @param xmlPayload the attribute values data for the row in the form of XML payload @return a boolean indicating the success of the operation @throws FatalException	This method publishes the given {@link BAMDataObject} to BAM server. It extracts Data Object Path, Comma Separated Key Fields as String and XML Payload from the given data object and invokes the BAM Data object operations web service through the corresponding adapter. @param bamDataObject the {@link BAMDataObject} to be published @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAalyticsWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\workflow\IAalyticsWebServicesAdapter	deleteRowFromBAMDataObject	This method will delete a row from the specified BAM Data Object. @param dataObject the fully qualified id of the BAM Data Object @param keysCSV the comma separated string of the key attributes of the specified BAM Data Object @param xmlPayload the attribute values data for the row in the form of XML payload @return a boolean indicating the success of the operation @throws FatalException	This method publishes the given {@link BAMDataObject} to BAM server. It extracts Data Object Path, Comma Separated Key Fields as String and XML Payload from the given data object and invokes the BAM Data object operations web service through the corresponding adapter. @param bamDataObject the {@link BAMDataObject} to be published @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makeAdvancePayment	The following method is used to do loan advance payment(deposit without reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to make advance payment.
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makeAdvancePayment	The following method is used to do loan advance payment(deposit without reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to execute the loan account related transactions based on the provided reason only when the account in context is not a loan account. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makeAdvancePayment	The following method is used to do loan advance payment(deposit without reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following is a common method which is called from various adapters when the account in context is a loan account. This method is used to various adapters to execute loan account related transaction. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makePartialPayOffLoan	The following method is used to do partial payoff (deposit with reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to perform partial payoff
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makePartialPayOffLoan	The following method is used to do partial payoff (deposit with reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to execute the loan account related transactions based on the provided reason only when the account in context is not a loan account. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	makePartialPayOffLoan	The following method is used to do partial payoff (deposit with reason) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following is a common method which is called from various adapters when the account in context is a loan account. This method is used to various adapters to execute loan account related transaction. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	closeLoanAccount	The following method is called when the reason is efs @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used when reason specified is EFS
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	closeLoanAccount	The following method is called when the reason is efs @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to execute the loan account related transactions based on the provided reason only when the account in context is not a loan account. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	closeLoanAccount	The following method is called when the reason is efs @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following is a common method which is called from various adapters when the account in context is a loan account. This method is used to various adapters to execute loan account related transaction. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	redrawFromLoanAccount	The following method is used to redraw from loan (withdraw) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to to redraw from loan account
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	redrawFromLoanAccount	The following method is used to redraw from loan (withdraw) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following method is used to execute the loan account related transactions based on the provided reason only when the account in context is not a loan account. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	redrawFromLoanAccount	The following method is used to redraw from loan (withdraw) @param sessionContext @param loanTransactionRequestDTO @return @throws FatalException	The following is a common method which is called from various adapters when the account in context is a loan account. This method is used to various adapters to execute loan account related transaction. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	fetchLoanAccountBalances	The following method is used to fetch loan account balance details @param sessionContext @param accountId @return @throws FatalException	
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	inquireFullPayOff	This method is used for Loan Full Pay Off Inquiry @param applicationContext @param accountId @param penaltyMethod @return LoanFullPayOffInquiryResponse	
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	inquireFullPayOff	This method is used for Loan Full Pay Off Inquiry @param applicationContext @param accountId @param penaltyMethod @return LoanFullPayOffInquiryResponse	The following method is used to execute the loan account related transactions based on the provided reason only when the account in context is not a loan account. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	inquireFullPayOff	This method is used for Loan Full Pay Off Inquiry @param applicationContext @param accountId @param penaltyMethod @return LoanFullPayOffInquiryResponse	The following is a common method which is called from various adapters when the account in context is a loan account. This method is used to execute loan account related transaction. @param businessContext @param loanTransactionRequestDTO @param reason @param isInflowTransaction @param contextResponse @param transactionType @return @throws FatalException
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	inquireFullPayOff	This method is used for Loan Full Pay Off Inquiry @param applicationContext @param accountId @param penaltyMethod @return LoanFullPayOffInquiryResponse	The following method is used to inquire about the loan account. It provides you necessary information that is needed before closing a loan account.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	createAccountClosureRequest	<p>This method is responsible for raising Account Closure Request for a Loan Account.  Once the request is raised, Loan Account status is changed to "Marked for Closure". </p> @param sessionContext @param loanAccountClosureRequestDTO @return	
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	fetchArrearsDueAmount	The following method is used to fetch the total arrear due amount. @param accountId @return @throws FatalException	The following method is used to fetch the balances for the account which is in context. Ballance display pattern is different for different account type's.
IDSLLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLLoanAdapter	fetchArrearsDueAmount	The following method is used to fetch the total arrear due amount. @param accountId @return @throws FatalException	
ICardMDMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardMDMAdapter	fetchMDMPartyDetails	Fetch MDM Party Details @param sessionContext @param internalPartyId @param externalPartyId @param partyType @return @throws FatalException	The method inquireCardAccountDetails is used to fetch credit card account details for the given card account number.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, ,This attribute denotes the card account number. @return {@link CreditCardAccountDetailsResponse} This class contains the information related to credit card account details. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardMDMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardMDMAAdapter	fetchMDMPartyDetails	Fetch MDM Party Details @param sessionContext @param internalPartyId @param externalPartyId @param partyType @return @throws FatalException	This method is used to fetch Credit Card Details for internal or external party. The party details and credit card reference number are fetched from MDM and using the credit card reference number, the card details are fetched from the credit card external system @param sessionContext @param partyId @param externalPartyId @return creditCardDetailsDTOs, CreditCardDetailsDTO[] @throws FatalException
ICardMDMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardMDMAAdapter	fetchMDMPartyDetails	Fetch MDM Party Details @param sessionContext @param internalPartyId @param externalPartyId @param partyType @return @throws FatalException	
ICardMDMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardMDMAAdapter	fetchMDMPartyDetails	Fetch MDM Party Details @param sessionContext @param internalPartyId @param externalPartyId @param partyType @return @throws FatalException	Construct party data DTO @param sessionContext @param partyId @param externalPartyId @param partyType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestAccountingAdapter	raiseAccountingEvent	This method is used to raise the accounting event. @param accountingRequest @throws FatalException	This method will raise Accounting Event that is related to Interest Processing. The method will internally delegate the call to {@link InterestAccountingAdapter} which will internally route the call to accounting module. This method accepts {@link AccountingRequest} object as input which contains details of accounting event. Following is the algorithm for this method:  <ol> <li>Populate {@link InterestAccountingEvent} object using the details provided in {@link AccountingRequest} object.</li> <li>Get concrete instance of {@link IInterestAccountingAdapter} object using Adapter Factory.</li> <li>Call raiseAccountingEvent method of {@link IInterestAccountingAdapter} to raise accounting event. Send {@link InterestAccountingEvent} object as input to this method.</li> <ol> @param accountingRequest {@link AccountingRequest} object containing details of accounting event.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestAccountingAdapter	raiseReverseAccountingEventForTransaction	This method is used to raise a reversal event for an entire transaction. @param transactionReferenceNumber @param originalTransactionReferenceNumber the transaction reference number of the transaction that needs to be reversed. @throws FatalException	This method will reverse all the raised accounting events that is related to Interest Processing. The method will internally delegate the call to {@link IDomainAccountingAdapter} which will internally route the call to accounting module. This method accepts two inputs: <ol><li>transactionReferenceNumber</li><li>originalTransactionReferenceNumber</li></ol>Following is the algorithm for this method: <ol><li>Get concrete instance of {@link IDomainAccountingAdapter} object using Adapter Factory.</li><li>Call raiseReverseAccountingEvent method of {@link IDomainAccountingAdapter} to reverse the accounting event for the originalTransactionReferenceNumber.</li></ol>@param transactionReferenceNumber This field holds the reference number against each transaction. This reference number is used to track a particular transaction. @param originalTransactionReferenceNumber This field holds the original transaction reference number. This reference number corresponds to the transaction that needs to be reversed. Using this reference number, the accounting events that were raised earlier will be reversed. @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	originateCreditCard	This method is used to originate credit card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	This method is used to originate credit card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	originateCreditCard	This method is used to originate credit card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	originateCreditCard	This method is used to originate credit card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference. After that it will save the current state of simulation. Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders, add-on card holders, account preferences such as balance transfer details, delivery preferences, card membership details and also the preferences for each card holder such as emboss name, card background image @return {@link CreditCardAccountResponse} This class contains account id and account configuration DTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination CreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	amendCreditCard	This method is used to amend Credit Card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	This method is used to amend Credit Card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException
IOrigination CreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	amendCreditCard	This method is used to amend Credit Card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	
IOrigination CreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	amendCreditCard	This method is used to amend Credit Card @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology.This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference.After that it will save the current state of simulation.Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders,add-on card holders, account preferences such as balance transfer details,delivery preferences,card membership details and also the preferences for each card holder such as emboss name,card background image @return (@link CreditCardAccountResponse) This class contains

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account id and account configuration DTO @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	isAccountOpened	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return Boolean @throws FatalException	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	isAccountOpened	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return Boolean @throws FatalException	
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	isAccountOpened	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return Boolean @throws FatalException	Validate if Credit card offer allowed for the applied party <ul> <li>Check if account already opened for the party id, offer id combination</li> <li>Check if any credit card application is in progress for the party id, offer id combination</li> </ul> @param sessionContext @param partyId @param offerId @return List<ValidationError> @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	isAccountOpened	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return Boolean @throws FatalException	Validate if Credit card offer allowed for the applied party <ul> <li>Check if account already opened for the party id, offer id combination</li> <li>Check if any credit card application is in progress for the party id, offer id combination</li> </ul> @param sessionContext @param partyId @param offerId @return List<ValidationError> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	isAccountOpened	This method is used to check is account opened for the party id, offer id combination @param sessionContext @param partyId @param offerId @return Boolean @throws FatalException	Validate if Credit card offer allowed for the applied party <ul> <li>Check if account already opened for the party id, offer id combination</li> <li>Check if any credit card application is in progress for the party id, offer id combination</li> </ul> @param sessionContext @param partyId @param offerId @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostOriginateCreditCardOperation	This method is used to perform post originate credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	This method is used to perform post originate credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostOriginateCreditCardOperation	This method is used to perform post originate credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostOriginateCreditCardOperation	This method is used to perform post originate credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	The method performPostOriginateOrAmendCreditCardOperation is used to perform post originate Credit Card operation.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders,add-on card holders, account preferences such as balance transfer details,delivery preferences,card membership details and also the preferences for each card holder such as emboss name,card background image @return {@link CreditCardAccountResponse} This class contains account id and account configuration DTO @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostAmendCreditCardOperation	This method is used to perform post amend credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	This method is used to perform post amend credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostAmendCreditCardOperation	This method is used to perform post amend credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	performPostAmendCreditCardOperation	This method is used to perform post amend credit card operation @param sessionContext @param creditCardAccountInputDTO @return CreditCardAccountResponse @throws FatalException	The method performPostOriginateOrAmendCreditCardOperation is used to perform post originate Credit Card operation.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders,add-on card holders, account preferences such as balance transfer details,delivery preferences,card membership details and also the preferences for each card holder such as emboss name,card background image @return {@link CreditCardAccountResponse} This class contains account id and account configuration DTO @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	inquireCreditCardDetailsByAccountNumber	This method is used to inquire credit card details by account number @param sessionContext @param accountNumber @return CreditCardDetailsDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	inquireCreditCardDetailsByAccountNumber	This method is used to inquire credit card details by account number @param sessionContext @param accountNumber @return CreditCardDetailsDTO @throws FatalException	This service will populate the simulation details for the product group for a given credit card application. It will iterate for all the product group linkages which are part of the application. For every Product offer references inside the product group linkage, it will fetch and stores the simulation if there is anything available. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardApplicationDTO, creditCardApplicationDTO, , Indicates the credit card application for which simulation details are being populated. @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	fetchBillCode	This method is used to fetch bill code for the Credit Card Account @param sessionContext @param cardAccountConfigurationDTO @param creditFacilityDTO @return CreditCardBillCodeResponse @throws FatalException	This method is used to fetch bill code for the Credit Card Account @param sessionContext @param cardAccountConfigurationDTO @param creditFacilityDTO @return CreditCardBillCodeResponse @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	fetchBillCode	This method is used to fetch bill code for the Credit Card Account @param sessionContext @param cardAccountConfigurationDTO @param creditFacilityDTO @return CreditCardBillCodeResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	linkAddOnCardHolder	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card @param sessionContext @param addOnCardHolderDTO @return TransactionStatus @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IOriginationCreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	linkAddOnCardHolder	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card @param sessionContext @param addOnCardHolderDTO @return TransactionStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination CreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	createDocumentChecklistForAddOnCardHolder	This service creates the inbound document checklist for the add-on card holder  @param sessionContext @param addOnCardHolderDTO @param creditCardDetailsDTO @param creditCardAccountConfigurationDTO @throws FatalException	This service creates the inbound document checklist for the add-on card holder  @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param AddOnCardHolderDTO, addOnCardHolderDTO, ,This class holds the information of add on card  holder for which inbound document checklist is to be created   @param CreditCardDetailsDTO, creditCardDetailsDTO, ,This class holds the information of credit card  to which addon card holder is being linked. This is optional argument when addon card holder is linked  during origination of credit card. In such case creditCardAccountConfigurationDTO parameter is required   @param CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, ,This class holds the information of credit card  account configuration. This is optional argument when addon card holder is added from credit card servicing screen  However in such case, creditCardDetailsDTO parameter is required  @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException  
IOrigination CreditCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCreditCardAdapter	createDocumentChecklistForAddOnCardHolder	This service creates the inbound document checklist for the add-on card holder  @param sessionContext @param addOnCardHolderDTO @param creditCardDetailsDTO @param creditCardAccountConfigurationDTO @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	isInterestRuleLinkedTo Product	This method checks whether the {@link InterestRule} send as input to it is attached to a product. If the interest rule is linked, it will return true, else it will return false. @param interestRule @return boolean indicating whether the interest rule is linked to a product or not.	A mistake such as data entry error may occur while creating an interest rule. In such cases, this method will be called to fetch and update the existing interest rule with the correct values. This method will not allow you to update values of the {@link InterestRuleKey}. Since every other field can be updated, all the mandatory checks, business policies and specifications will be validated.   This method is called to update a base interest rule. The update works as follows: <ol> <li>It will first check whether the interest rule has already been linked to a product.   <b>Once linked to a product, update to that rule is not allowed.</b>  {@link ProductManufacturingService} is called to perform the required check, .</li> <li>This method allowed updation of all fields other than the {@link InterestRuleKey}.</li> <li>It performs a check on the mandatory fields of the interest rule.</li> <li>It then verifies the set up of the interest global parameters. i.e the input combination of {@link ProductClass}, {@link InterestClass}, {@link InterestDivision} and {@link InterestConstituentType}.</li> <li>Further the business policy and specifications are validated.</li> </ol> @param interestRule the interest rule to be updated.
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	isInterestRuleLinkedTo Product	This method checks whether the {@link InterestRule} send as input to it is attached to a product. If the interest rule is linked, it will return true, else it will return false. @param interestRule @return boolean indicating whether the interest rule is linked to a product or not.	This method is used to close an interest rule.   The method works as follows: <ul> <li>Accept the domain object InterestRule as an input parameter.</li> <li>Call the method {@link InterestRuleRepository#getInstance} to get an unique instance.</li> <li>Call the close method of the base class MaintenanceDomainService as super.close(interestRule, interestRuleRepository)</li> </ul> @param interestRule {@link InterestRule}.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	isInterestRuleLinkedToProduct	This method checks whether the {@link InterestRule} send as input to it is attached to a product. If the interest rule is linked, it will return true, else it will return false. @param interestRule @return boolean indicating whether the interest rule is linked to a product or not.	This method checks whether the {@link InterestRule} send as input to it is attached to a product. If the interest rule is linked, it will return true, else it will return false. @param interestRule @return boolean indicating whether the interest rule is linked to a product or not.
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method is called to fetch the {@link InterestRule} which is linked to the product.  For Interest calculation purpose there are certain parameters defined in {@link InterestRule} which are required .  The parameters which are required for Interest processing are as follows :-   <ul> <li> {@link DayCountConventionType} </li> {@link RoundingRule} </ul> @param productId @return InterestRule
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response.  @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method fetches the IRD linked to the input product Id , {@link InterestClass} and {@link InterestDivision} combination.This method fetches all the IRD linkages for the input productId and returns the IRD specified for the input {@link InterestClass} and {@link InterestDivision}. If there is no IRD maintained for the input, thenThis method will return null. @param productId @param interestClass @param interestDivision @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	getInterestRules	This method fetches interest rule definition by calling <code>fetchInterestRuleDefinition</code> from <code>InterestRuleDefinitionApplicationService</code> and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of <a href="#">InterestRuleKeyDTO</a> objects which hold interest rule key parameters.	This method is called to fetch the <a href="#">InterestRule</a> which is linked to the product , <a href="#">InterestDivision</a> } and <a href="#">InterestClass</a> combination.  For Interest calculation purpose the following parameters defined in <a href="#">InterestRule</a> are required  <li> <a href="#">DayCountConventionType</a> </li> <li>Capitalization <a href="#">RoundingRule</a> </li> <li> <a href="#">BaseForInterest</a> for the input <a href="#">LoansInterestSimulationRequest#interestConstituentType</a> </li> </ol> <ol> <li>Create an instance of <a href="#">IInterestPMUAdapter</a> </li> <li>Call <a href="#">IInterestPMUAdapter#getInterestRules</a> and receive array of <a href="#">InterestRuleKeyDTO</a> </li> <li>Iterate over the array and get the interest rule key DTO with <code>(<a href="#">InterestRuleKeyDTO#interestDivision</a> == <a href="#">interestDivision</a>) &amp;&amp; (<a href="#">InterestRuleKeyDTO#interestClass</a> == <a href="#">interestClass</a>)</code> </li> <li>Create an object of <a href="#">InterestRuleKey</a> and set the attributes from the <a href="#">InterestRuleKeyDTO</a> object </li> <li>Load the interest rule using <a href="#">InterestRuleResolutionService.getInstance().fetchInterestRule</a> and return the interest rule.</li> </ol> @param productId @param interestClass @param interestDivision @return
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	getInterestRules	This method fetches interest rule definition by calling <code>fetchInterestRuleDefinition</code> from <code>InterestRuleDefinitionApplicationService</code> and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of <a href="#">InterestRuleKeyDTO</a> objects	This method fetches interest rule definition by calling <code>fetchInterestRuleDefinition</code> from <code>InterestRuleDefinitionApplicationService</code> and populates interest rule key DTO from the response.  @param productId The product for which interest rule definition need to be fetched. @return Array of <a href="#">InterestRuleKeyDTO</a> objects which hold interest rule key parameters.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			which hold interest rule key parameters.	
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	fetchProduct	This method will fetch the product details. @param productCode @return	This method will fetch the TD product redemption parameters. @param productCode @return
IInterestPMU Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPMU Adapter	fetchProduct	This method will fetch the product details. @param productCode @return	This method will fetch the product details. @param productCode @return
IReportsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\reports\IReportsContentAdapter	createDocument	/ create the document in the Document Management system. @param sessionContext the session context @param branchCode the branch code @param customerId the customer id @param accountNumber the account number @return newDocumentId	Create the document in the Document Management system and return the document id of the newly created document. @param sessionContext @param id @param isBranchCode @return String
IReportsContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\reports\IReportsContentAdapter	createDocument	/ create the document in the Document Management system. @param sessionContext the session context @param branchCode the branch code @param customerId the customer id @param accountNumber the account number @return newDocumentId	/ create the new document in the Document Management system and returns the document id of the created document if isBranchCode flag is true then index would be branch code else it would be customer id. @param sessionContext @param customerId
IContactPointSequenceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IContactPointSequenceAdapter	fetchContactPointSequence		The following method is used to populate the contact points. @param sessionContext @param partyId @param partyDTO @param contactPointAppService @param contactPointResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method will be used for populating the activity log and logging the alert for all the maintenance transactions. @param activityId @param eventId @param accountId @param object @param sessionContext @throws FatalException
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method will be used for populating the activity log and logging the alert for all the maintenance transactions. @param activityId @param eventId @param accountId @param object @param sessionContext @throws FatalException
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	@param activityId @param eventId @param accountId @param newObject @param oldObject
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method persists activity and event log for given activity and event id @param activityId @param eventId @param activityLog @param eventDate @throws FatalException
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method will be used for populating the activity log and logging the alert for all the maintenance transactions. @param activityId @param eventId @param accountI @param object @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method will be used for populating the activity log for modification events where both old and new data is required by the alert messages. @param activityId @param eventId @param accountId @param newObject @param oldObject @throws FatalException
IDDAEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method persists activity and event log for given activity and event id @param activityId @param eventId @param activityLog @param eventDate @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	inquireInterestAccruedTillDate	This method calls method from Interest inquiry service to inquire details of interest for term deposit which includes unrecovered penalty amount. @param arrangementNumber Containing account and deposit id @param dueDate Due date for the action @return {@link TDInterestInquiryResponse} Object containing interest inquiry details	This method calls method from Interest inquiry service to inquire details of interest for term deposit which includes unrecovered penalty amount. @param arrangementNumber Containing account and deposit id @param dueDate Due date for the action @return {@link TDInterestInquiryResponse} Object containing interest inquiry details.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	inquireInterestAccruedTillDate	This method calls method from Interest inquiry service to inquire details of interest for term deposit which includes unrecovered penalty amount. @param arrangementNumber Containing account and deposit id @param dueDate Due date for the action @return {@link TDInterestInquiryResponse} Object containing interest inquiry details	This method is responsible for the processing of deposit balance change Action   Interest computation happens on deposit balance change This method would be invoked to process actions during end of day for the account whose action details is passed in input.   The main job of This method is to accept the inputs and invoke {@link ITDInterestAdapter#performInterestComputation()} with the same.   After the TDActionLogger {@link TDActionLogger#logTDBalanceChangeAction()} is called to log the deposit balance change event in TD.</br> Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} Object containing account and deposit id @param actionRequest {@link TDActionRequest} Object containing td action to be performed @param dueDate Due date for the action
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	inquireRedemptionDetails	This method will fetch Interest details from interest module. These interest details are required during Term Deposit Redemption process. @param redemptionInterestInquiryRequest {@link RedemptionInterestInquiryRequestDTO} Object containing details required for interest inquiry @return {@link RedemptionInterestInquiryResponseDTO} Object containing interest details required during Term Deposit Redemption	This method will fetch Interest details from interest module.  These interest details are required during Term Deposit Redemption process. @param redemptionInterestInquiryRequestDTO {@link RedemptionInterestInquiryRequestDTO} Object containing details required for interest inquiry @return {@link RedemptionInterestInquiryResponseDTO} Object containing interest details required during Term Deposit Redemption @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#inquireRedemptionDetails(com.ofss.fc.app.adapter.td.dto.RedemptionInterestInquiryRequestDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	inquireRedemptionDetails	This method will fetch Interest details from interest module. These interest details are required during Term Deposit Redemption process. @param redemptionInterestInquiryRequest {@link RedemptionInterestInquiryRequest DTO} Object containing details required for interest inquiry @return {@link RedemptionInterestInquiryResponseDTO} Object containing interest details required during Term Deposit Redemption	This method interacts with interest module and gets interest details. @param deposit Term deposit details @param adminFee Amount charged by bank @param economicCost Economic cost generated by system or entered by OE @param feeRecoverySequence {@link FeeRecoverySequence} The sequence for fee recovery @param redemptionDate Date on which redemption done @param isPostMaturityRedemption Flag to check whether the redemption type is post maturity @param redemptionType {@link RedemptionType} Type of redemption @return {@link RedemptionInterestInquiryResponseDTO} Object which contains details of interest inquiry
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	inquireRedemptionDetails	This method will fetch Interest details from interest module. These interest details are required during Term Deposit Redemption process. @param redemptionInterestInquiryRequest {@link RedemptionInterestInquiryRequest DTO} Object containing details required for interest inquiry @return {@link RedemptionInterestInquiryResponseDTO} Object containing interest details required during Term Deposit Redemption	This method inquires redemption details and will also calculate the maturity outstanding penalty amount, tax applicable and net interest payable on remaining deposit after redemption. @param redemptionRequestDTO {@link RedemptionRequestDTO} Object containing details required for premature redeem TD simulation @return {@link PrematureRedeemTDDTO} Object containing redemption details and maturity, interest details in case of premature part redeem simulation



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doRedemptionInterestProcessing	This method will fetch interest details and processes interest for redemption. @param redemptionInterestProcessingRequest {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest processing @return {@link RedemptionInterestProcessingResponseDTO} Object containing interest details required for redemption	This method will fetch interest details and processes interest for redemption. @param redemptionInterestProcessingRequest {@link RedemptionInterestProcessingRequestDTO} object containing details required for interest processing @return {@link RedemptionInterestProcessingResponseDTO} object containing interest details required for redemption @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#doRedemptionInterestProcessing(com.ofss.fc.app.adapter.td.dto.RedemptionInterestProcessingRequestDTO)
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doRedemptionInterestProcessing	This method will fetch interest details and processes interest for redemption. @param redemptionInterestProcessingRequest {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest processing @return {@link RedemptionInterestProcessingResponseDTO} Object containing interest details required for redemption	This method interacts with interest module for interest processing at the time of actual redemption. @param deposit Term deposit details @param adminFee Amount charged by bank @param economicCost Economic cost generated by system or entered by OE @param feeRecoverySequence {@link FeeRecoverySequence} The sequence for fee recovery @param redemptionDate Date on which redemption done @param redemptionType {@link RedemptionType} Type of redemption @param transactionRefNo Transaction reference number @return {@link RedemptionInterestProcessingResponseDTO} Object containing information of interest processing before actual redemption

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateUnrecoveredFeesInInterest	This method will notify and update about the unrecovered fees to the interest module.  If fee amount has been totally recovered from interest, unrecovered fees will be 0.  @param redemptionInterestProcessingRequestDTO {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest inquiry or interest processing	This method will notify and update about the unrecovered fees to the interest module.  If fee amount has been totally recovered from interest, unrecovered fees will be 0.  @param redemptionInterestProcessingRequestDTO {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest inquiry or interest processing @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#updateUnrecoveredFeesInInterest(com.ofss.fc.app.adapter.td.dto.RedemptionInterestProcessingRequestDTO)
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateUnrecoveredFeesInInterest	This method will notify and update about the unrecovered fees to the interest module.  If fee amount has been totally recovered from interest, unrecovered fees will be 0.  @param redemptionInterestProcessingRequestDTO {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest inquiry or interest processing	This method updates the interest for unrecovered fees after recovering penalty @param deposit Term deposit details @param adminFee Amount charged by bank @param economicCost Economic cost generated by system or entered by OE @param feeRecoverySequence {@link FeeRecoverySequence} The sequence for fee recovery @param redemptionDate Date on which redemption done @param redemptionType {@link RedemptionType} Type of redemption
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateUnrecoveredFeesInInterest	This method will notify and update about the unrecovered fees to the interest module.  If fee amount has been totally recovered from interest, unrecovered fees will be 0.  @param redemptionInterestProcessingRequestDTO {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest inquiry or	This method recovers the outstanding penalty from the principal balance of deposit if waive outstanding penalty at maturity is false.   It also raises the accounting events for outstanding admin fee and economic cost which is recovered from principal balance.  @param maturityProcessInformation {@link MaturityProcessInformation}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interest processing	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateUnrecoveredFeeInInterest	This method will notify and update about the unrecovered fees to the interest module.  If fee amount has been totally recovered from interest, unrecovered fees will be 0.  @param redemptionInterestProcessingRequestDTO {@link RedemptionInterestProcessingRequestDTO} Object containing details required for interest inquiry or interest processing	This method updates the interest for unrecovered fees after recovering penalty @param deposit Term deposit details @param adminFee Amount charged by bank @param economicCost Economic cost generated by system or entered by OE @param feeRecoverySequence {@link FeeRecoverySequence} The sequence for fee recovery @param redemptionDate Date on which redemption done @param redemptionType {@link RedemptionType} Type of redemption
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	performInterestComputation	This method is used to perform send interest computation request for particular term deposit.  Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return	Based on the flag isRedemptionRenewalforUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePrincipalDue and UncollectedPrincipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	performInterestComputation	This method is used to perform send interest computation request for particular term deposit.  Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return	This method is used to perform send interest computation request for particular term deposit.  Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	performInterestComputation	This method is used to perform send interest computation request for particular term deposit. Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return	This method uses adapter for interest capitalization and also logs TDFinancialItemStatement. @param actionRequest @param accountId @param depositId @param dueDate @param maturityDate TODO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	performInterestComputation	This method is used to perform send interest computation request for particular term deposit. Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return	This method is responsible for the processing of deposit balance change Action   Interest computation happens on deposit balance change This method would be invoked to process actions during end of day for the account whose action details is passed in input.   The main job of This method is to accept the inputs and invoke {@link ITDInterestAdapter#performInterestComputation()} with the same.   After the TDActionLogger {@link TDActionLogger#logTDBalanceChangeAction()} is called to log the deposit balance change event in TD.  Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} Object containing account and deposit id @param actionRequest {@link TDActionRequest} Object containing td action to be performed @param dueDate Due date for the action

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	performInterestComputation	This method is used to perform send interest computation request for particular term deposit. Interest module will perform actual interest computation. Next Capitalization date for the applicable Interest Division will be returned @param accountId @param depositId @param interestDivisionDTO @return	This method marks the deposit as mature. The Balances are updated according to the Principal Payout Option. if {@link PrincipalPayoutOption#NO_INSTRUCTION} is set then :  <ol> <li>1. Principal Balances are moved to Uncollected Principal Balance. {@link Balance#uncollectedPrincipalBalance} += {@link Balance#principleBalance} + {@link Balance#BalancePrincipalDue}</li> <li>Interest Balances are moved to Uncollected Interest. {@link Balance#uncollectedInterestBalance} += {@link Balance#BalanceInterestDue}</li> <li>The following buckets are zeroised : {@link Balance#principleBalance}, {@link Balance#BalancePrincipalDue}, {@link Balance#BalanceInterestDue}</li> </ol> <p> If {@link PrincipalPayoutOption#RE_DEEM} is set then :  <li>1. Principal Balances are moved to Principal Balance Due. {@link Balance#balancePrincipalDue} += {@link Balance#principleBalance} + {@link Balance#uncollectedPrincipalBalance}</li> <li>Interest Balances are moved to Interest Balance Due. {@link Balance#balanceInterestDue} += {@link Balance#uncollectedInterestBalance}</li> <li>The following buckets are zeroised : {@link Balance#principleBalance}, {@link Balance#uncollectedPrincipalBalance}, {@link Balance#uncollectedInterestBalance}</li> </ol> </p> @param deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	registerInterestPlan	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module. @param tdInterestRegistrationDTO {@link TDInterestRegistrationDTO} Object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Containing result of registration process @throws FatalException	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module. @param tdInterestRegistrationDTO {@link TDInterestRegistrationDTO} Object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Containing result of registration process @throws FatalException @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#registerInterestPlan(com.ofss.fc.app.adapter.td.dto.TDInterestRegistrationDTO)
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	registerInterestPlan	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module. @param tdInterestRegistrationDTO {@link TDInterestRegistrationDTO} Object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Containing result of registration process @throws FatalException	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	registerInterestPlan	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module. @param tdInterestRegistrationDTO {@link TDInterestRegistrationDTO} Object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Containing result of registration process @throws FatalException	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module.  It also logs the payout action in TDActionLogger. @param deposit {@link DepositDTO} object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Outcome of registration.
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateArrangementStatus	@param accountId the identifier of the account. @param depositId the deposit for which the deposit status is to be changed. @param depositStatus the status for the interest	@param accountId the identifier of the account. @param depositId the deposit for which the deposit status is to be changed. @param depositStatus the status for the interest
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateArrangementStatus	@param accountId the identifier of the account. @param depositId the deposit for which the deposit status is to be changed. @param depositStatus the status for the interest	This is meant to change the depositStatus of a deposit to regular from fundAwaited, If the isSuccess comes true else close the deposit. @param accountId {@link String} holds the account Id under which the deposit is present. @param depositId {@link Integer} holds the deposit id of the deposit. @param isSuccess {@link Boolean} the flag from settlement, which will tell us to change the deposit status. @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	unRegisterInterestPlan	This method will be used by payinReversal to unregister the plan @param accountId Term deposit account id @param depositID Term deposit id	This method will be used by payinReversal to unregister the plan @param accountId term deposit account id @param depositID term deposit id @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#unRegisterInterestPlan(java.lang.String, java.lang.Integer)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	unRegisterInterestPlan	This method will be used by payinReversal to unregister the plan @param accountId Term deposit account id @param depositID Term deposit id	This method reverses the paying transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed. @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction. @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @param subSequenceNo TODO
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	unRegisterInterestPlan	This method will be used by payinReversal to unregister the plan @param accountId Term deposit account id @param depositID Term deposit id	This method will close the deposit @param deposit {@link Deposit} deposit to be closed @param closureDate Date on which deposit is being closed
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	unRegisterInterestPlan	This method will be used by payinReversal to unregister the plan @param accountId Term deposit account id @param depositID Term deposit id	The method de-register the deposit from Interest module. @param deposit {@link Deposit} The term deposit for unregister
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	reverseRedemption	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed.	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed. @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#reverseRedemption(com.ofss.fc.datatype.Date, java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	reverseRedemption	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed.	This method reverses the redemption transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	reverseRedemption	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed.	This method reverses the renewal transaction @param originalTransactionReferenceNumber {@link String} Holds the original transactionReferenceNumber that is used to identify the transaction to be reversed.  @param internalTransactionReferenceNumber {@link String} Holds the transactionReferenceNumber of the current transaction.  @param isAccountinReversalRequired informs that the accounting is to raised @param taskCode Identifier of the transaction to be reversed @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	reverseRedemption	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed.	The method calls the tax module to reverse the sequence of operations done in tax due to redemption.  @param internalReferenceNumber Holds the transactionReferenceNumber of the current transaction. @param originalTransactionReferenceNumber Holds the transactionReferenceNumber of the transaction, which is meant to be reversed.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	reverseRedemption	On reversal of the redemption reversal, this method will come in execution. This method notifies interest module that the transaction of redemption is reversed. @param lastTransactiondate Date of last transaction @param originalTransactionReferenceNo Reference number of the transaction meant to be reversed.	The method calls the interest module to reverse the sequence of operations done in interest due to redemption. The method also take care of the back dated interest reversal also if the redemption was premature full. @param deposit {@link Deposit} holds the attributes of deposit entity @param lastTransactiondate Holds the date on which the last transaction was performed @param originalTransactionReferenceNumber Holds the transactionReferenceNumber of the transaction, that is meant to be reversed. @param redemptionType {@link RedemptionType} tells the type of redemption
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the compounded interest on a deposit @param depositDTO @return Money

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method will provide the Collateral Details for the given deposit @param sessionContext @param depositDTO @return TDCollateralDTO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method will fetch the collateral liens placed on all Accounts for the given party id. External service/module will call this method to fetch the collateral values for all account and deposits by giving party ids as input. @param sessionContext @param partyIds @param collateralId @return TDCollateralResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the Interest related details of deposit @param depositDTO
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#fetchDepositInterestPlanDetails(java.lang.String, java.lang.Integer, com.ofss.fc.datatype.Date, com.ofss.fc.enumeration.interest.InterestDivision)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the compounded interest till date for a deposit. @param deposit @return Money
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the compounded interest till date for a deposit. @param deposit @return Money

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the compounded interest till date for a deposit. @param deposit @return Money
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method generates the {@link DepositAttributeDTO} from the {@link DepositDetailInquiry} provided by {@link DepositSummaryView} from domain service {@link DepositInquiryService} @param depositDetailInquiry

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	<p>This method will give Compounding parameter of the deposit.&lt;br&gt; This may include compounding frequency,next compounding date,previous compounding date etc&lt;br&gt; @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item</p>	<p>This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process&lt;/li&gt;</li> <li>&lt;li&gt;Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information</li> <li>&lt;li&gt;The {@link RenewalResponse} containing any amount which has to be renewed&lt;/li&gt;</li> <li>&lt;li&gt;The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}&lt;/li&gt;</li> <li>&lt;li&gt;The Deposit which is being Matured&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; &lt;/li&gt; &lt;/li&gt; &lt;li&gt;The method reads the RenewalResponse. &lt;p&gt; If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances &lt;/p&gt; &lt;p&gt; If Deposit Returned from Renewal is null then {@link DepositMaturityService#matureDeposit} is invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	<p>This method will give Compounding parameter of the deposit.&lt;br&gt; This may include compounding frequency,next compounding date,previous compounding date etc&lt;br&gt; @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item</p>	<p>This method fetches the Interest related details of deposit @param depositAttributeInquiryDTO @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the Compounded Interest Frequency @param deposit {@link Deposit} Term deposit for which interest compounding frequency needs to be fetched @return {@link FrequencyInterestCompoundingFrequency}
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the Interest related details of deposit @param depositDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method will fetch the interest plan item at a given instance in time. @param accountId @param depositNumber @param valueDate @param interestDivision @return
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method fetches the compounded interest till date for a deposit. @param deposit @return Money

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchDepositInterestPlanDetails	This method will give Compounding parameter of the deposit.  This may include compounding frequency,next compounding date,previous compounding date etc  @param accountId Term deposit account id @param depositId Term deposit id @param requestDate Date for records which are after this date @param interestDivision {@link InterestDivision} Enumeration which specifies interest rules @return {@link InterestPlanItemDetailsResponse} Object containing details of interest plan item	This method inquires redemption details and will also calculate the maturity outstanding penalty amount, tax applicable and net interest payable on remaining deposit after redemption. @param redemptionRequestDTO {@link RedemptionRequestDTO} Object containing details required for premature redeem TD simulation @return {@link PrematureRedeemTDDTO} Object containing redemption details and maturity, interest details in case of premature part redeem simulation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestEvents	This method will give the details of interest inquiry with event for the transaction. @param sessionContext The basic information about the user of the system. @param accountId Term deposit account id @param depositNumber Term deposit id @param startDate Event occurred starting from this date @param endDate Event occurred before this date @return List {@link InterestInquiryDTO} List of DTOs containing details of interest inquiry	This method will give the details of interest inquiry with event for the transaction. @param sessionContext The basic information about the user of the system. @param accountId Term deposit account id @param depositNumber Term deposit id @param startDate Event occurred starting from this date @param endDate Event occurred before this date @return List {@link InterestInquiryDTO} List of DTOs containing details of interest inquiry @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#getInterestEvents(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.Integer, com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestEvents	This method will give the details of interest inquiry with event for the transaction. @param sessionContext The basic information about the user of the system. @param accountId Term deposit account id @param depositNumber Term deposit id @param startDate Event occurred starting from this date @param endDate Event occurred before this date @return List {@link InterestInquiryDTO} List of DTOs containing details of interest inquiry	This method generates the list of {@link InterestDetailInquiryDTO} object from the {@link InterestEventDTO} provided by the external module service InterestInquiryApplicationService @param sessionContext @param accountId @param depositNumber @param startDate @param endDate @param transactionType @return List of InterestInquiryDTO
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doInterestSettlement	This method will be used to fill the TermDepositInterestSettlementResponseDTO. @param termDepositInterestSettlementRequestDTO {@link TermDepositInterestSettlementRequestDTO} Object containing details required for settlement of interest @return {@link TermDepositInterestSettlementResponseDTO} Object containing information after settlement of interest	This method will make a call to {@link TDInterestAdapter#fetchDepositInterestDetails} to fetch the Interest Capitalization, Accrued and Compounded till now for the Deposit.    It will pass the DepositNumber and DepositmaturityDate to the Adapter to get an {@link InterestPayoutDetailsDTO} object  @param accountId id of term deposit account @param depositNumber term deposit id @param dueDate due date for the action @param applicationContext Application Context which will be set in batch process. @return {@link TermDepositInterestSettlementResponseDTO} object containing pre and post tax amounts

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	doInterestSettlement	This method will be used to fill the TermDepositInterestSettlementResponseDTO. @param termDepositInterestSettlementRequestDTO {@link TermDepositInterestSettlementRequestDTO} Object containing details required for settlement of interest @return {@link TermDepositInterestSettlementResponseDTO} Object containing information after settlement of interest	This method will be used to fill the TermDepositInterestSettlementResponseDTO. @param termDepositInterestSettlementRequestDTO {@link TermDepositInterestSettlementRequestDTO} Object containing details required for settlement of interest @return {@link TermDepositInterestSettlementResponseDTO} Object containing information after settlement of interest @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#doInterestSettlement(com.ofss.fc.app.adapter.td.dto.TermDepositInterestSettlementRequestDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	doInterestCapitalization	This method will in turn call doInterestCapitalization() in InterestCapitalizationService.It prepares the interest rule and fetches next capitalization date. @ <a href="#">param dueDate</a> Date upto which interest processing needs to be done @ <a href="#">param interestClass</a> { <a href="#">@link InterestClass</a> } Higher level classification of interest rule @ <a href="#">param interestDivision</a> { <a href="#">@link InterestDivision</a> } Lower level classification of interest rule @ <a href="#">param isArrangementBeingClosed</a> Flag denotes arrangement closed or not @ <a href="#">param transactionReferenceNumber</a> Reference Number of the Transaction @ <a href="#">param maturityDate</a> TODO @ <a href="#">param arrangementNumber</a> String having accountId and depositId concatenated @ <a href="#">return</a> { <a href="#">@link InterestCapitalizationResponseDTO</a> } Object containing details returned from interest capitalization process	This method will in turn call doInterestCapitalization() in InterestCapitalizationService.It prepares the interest rule and fetches next capitalization date. @ <a href="#">param dueDate</a> Date upto which interest processing needs to be done @ <a href="#">param interestClass</a> { <a href="#">@link InterestClass</a> } Higher level classification of interest rule @ <a href="#">param interestDivision</a> { <a href="#">@link InterestDivision</a> } Lower level classification of interest rule @ <a href="#">param isArrangementBeingClosed</a> Flag denotes arrangement closed or not @ <a href="#">param transactionReferenceNumber</a> Reference Number of the Transaction @ <a href="#">param arrangementNumber</a> String having accountId and depositId concatenated @ <a href="#">return</a> { <a href="#">@link InterestCapitalizationResponseDTO</a> } Object containing details returned from interest capitalization process @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#doInterestCapitalization(com.ofss.fc.datatype.Date, com.ofss.fc.enumeration.interest.InterestClass, com.ofss.fc.enumeration.interest.InterestDivision, java.lang.String, int, com.ofss.fc.datatype.Date, boolean)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	doInterestCapitalization	<p>This method will in turn call doInterestCapitalization() in InterestCapitalizationService.It prepares the interest rule and fetches next capitalization date.</p> <p>@param dueDate Date upto which interest processing needs to be done</p> <p>@param interestClass {@link InterestClass} Higher level classification of interest rule</p> <p>@param interestDivision {@link InterestDivision} Lower level classification of interest rule</p> <p>@param isArrangementBeingClosed Flag denotes arrangement closed or not</p> <p>@param transactionReferenceNumber Reference Number of the Transaction</p> <p>@param maturityDate TODO</p> <p>@param arrangementNumber String having accountId and depositId concatenated</p> <p>@return {@link InterestCapitalizationResponseDTO} Object containing details returned from interest capitalization process</p>	<p>This method uses adapter for interest capitalization and also logs TDFinancialItemStatement.</p> <p>@param actionRequest</p> <p>@param accountId</p> <p>@param depositId</p> <p>@param dueDate</p> <p>@param maturityDate TODO</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method validates interest payout frequency
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#fetchNextInterestPhaseDate(com.ofss.fc.datatype.Date, com.ofss.fc.enumeration.td.Frequency, java.lang.String, com.ofss.fc.app.interest.dto.interestrule.InterestRuleKe



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interestCapitalization date	yDTO, com.ofss.fc.enumeration.interest.InterestPhase)
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhase Date	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Drequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method validates interest payout frequency
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Drequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or	This method consists of the logic to calculate Maturity amount and total interest earned. @param investmentAmount Initial amount @param depositTerm Term for deposit @param depositDate Date on which investment amount deposited @param maturityDate Date on which deposit will mature @param interestRate Net rate applicable for the amount and term @param payoutFrequency Payout frequency for TD @param compoundingFrequency Compounding frequency of TD @param maturityFactor {@link MaturityFactor} Specifies whether depositTerm or maturirty date present in input parameters @param productId TD product code @return {@link InvestmentAvenue} Object containing amount and term details after simulation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interestCapitalization date	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhase Date	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase { <a href="#">@link InterestPhase</a> } Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method updates the PayoutFrequency In Interest Plan if there is change in the frequency. @param deposit { <a href="#">@link Deposit</a> } Term deposit details after contract modification @param existDeposit { <a href="#">@link Deposit</a> } Deposit before contract modification

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Drequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method calculates and sends response to the screen for the display fields. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing details for computation @return {@link ContractDetailComputationResponse} Object containing response of computation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Drequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or	This method is used to fetch the Read only field during contract modification. The read only fields are as follows:- <ul style="list-style-type: none"> <li>Base Interest</li> <li>Margin</li> <li>UDM</li> <li>Net Interest Rate</li> <li>Interest Compounding Frequency</li> <li>Maturity Amont</li> </ul> @param depositContractModificationDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interestCapitalization date	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchNextInterestPhaseDate	This method is used to fetch first interest compounding date and interestCapitalization date. @param interestStartDate Date from which interest being calculated @param interestFrequency Frequency of interest depending on phase of interest @param productId Term deposit product id @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @param interestPhase {@link InterestPhase} Phase of interest @return Date derived first interest compounding date or interestCapitalization date	This method performs calculation of maturity amount and interest for the remaining balance after part redemption. @param remainingPrincipalBalance {@link Money} the principal balance after part redemption @param remainingTerm {@link Term} remaining term after redemption @param redemptionDate {@link Date} date on which redemption to be performed @param maturityDate {@link Date} date on which deposit will mature @param interestRate {@link BigDecimal} net interest rate applicable @param interestPayoutFrequency {@link Frequency} interest payout frequency of term deposit @param interestCompoundingFrequency {@link Frequency} interest compounding frequency of TD @param deposit {@link Deposit} Object containing term deposit details @return {@link InvestmentAvenue} Object containing amount and term details after simulation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doBackDatedInterestProcessing	This method does backdated interest processing for term deposit which are redeemed in minimum interest term. @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing term deposit facts required for bankingFactContext @param tdBackDatedInterestProcessingRequestDTO {@link TDBackDatedInterestProcessingRequestDTO} Object containing details required for backdated interest processing	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doBackDatedInterestProcessing	This method does backdated interest processing for term deposit which are redeemed in minimum interest term. @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing term deposit facts required for bankingFactContext @param tdBackDatedInterestProcessingRequestDTO {@link TDBackDatedInterestProcessingRequestDTO} Object containing details required for backdated interest processing	The method calls the interest module to reverse the sequence of operations done in interest due to redemption. The method also take care of the backdated interest reversal also if the redemption was premature full. @param deposit {@link Deposit} holds the attributes of deposit entity @param lastTransactiondate Holds the date on which the last transaction was performed @param originalTransactionReferenceNumber Holds the transactionReferenceNumber of the transaction, that is meant to be reversed. @param redemptionType {@link RedemptionType} tells the type of redemption
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doBackDatedInterestProcessing	This method does backdated interest processing for term deposit which are redeemed in minimum interest term. @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing term deposit facts required for bankingFactContext @param tdBackDatedInterestProcessingRequestDTO {@link TDBackDatedInterestProcessingRequestDTO} Object containing details required for backdated interest processing	This method performs a redemption. @param deposit Term deposit details @param redemptionTransactionDTO Details of redemption transaction @return {@link RedemptionResponseDetailsDTO} Object containing details of redemption after actual redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#doRedemption(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.app.td.dto.transaction.redemption.RedemptionTransactionDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	doBackDatedInterestProcessing	This method does backdated interest processing for term deposit which are redeemed in minimum interest term. @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing term deposit facts required for bankingFactContext @param tdBackDatedInterestProcessingRequestDTO {@link TDBackDatedInterestProcessingRequestDTO} Object containing details required for backdated interest processing	This method is responsible for the processing back dated interest processing Action  This method would be invoked to process actions during end of day for the account whose action details is passed in input.   The main job of this method is to accept the inputs and invoke {@link ITDInterestAdapter#doBackDatedInterestProcessing()} with the same.   After the TDActionLogger {@link TDActionLogger#logTDBackdatedInterestComputationAction()} is called to log the backdated interest processing event in TD.  Exceptions thrown by this method and the subsequent service calls are handled by the Batch Framework. @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} Object containing account and deposit id @param actionRequest {@link TDActionRequest} Object containing td action to be performed @param dueDate Due date for the action
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link InterestFrequency} InterestFrequency required to calculate interest payout and compounding dates	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link InterestFrequency} InterestFrequency required to calculate interest payout and compounding dates @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#getCompoundingFrequency(com.ofss.fc.app.interest.dto.interestrule.InterestRuleKeyDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link Frequency} InterestFrequency required to calculate interest payout and compounding dates	This method consists of the logic to calculate Maturity amount and total interest earned. @param investmentAmount Initial amount @param depositTerm Term for deposit @param depositDate Date on which investment amount deposited @param maturityDate Date on which deposit will mature @param interestRate Net rate applicable for the amount and term @param payoutFrequency Payout frequency for TD @param compoundingFrequency Compounding frequency of TD @param maturityFactor {@link MaturityFactor} Specifies whether depositTerm or maturity date present in input parameters @param productId TD product code @return {@link InvestmentAvenue} Object containing amount and term details after simulation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link Frequency} InterestFrequency required to calculate interest payout and compounding dates	The method returns the Interest Compounding Frequency which is present at interest definition rule. @param productId @return {@link Frequency}



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @return { <a href="#">@link Frequency</a> } InterestFrequency required to calculate interest payout and compounding dates	This method validates the pay-out frequency for the deposit against the product.  The new pay out frequency must adhere to the minimum interest pay out frequency maintained at the product level.   Such frequency might have been modified at the deposit level, check for the frequency at product level and permit modification up-to such frequency.   <b>For e.g. </b>  At product level - Interest pay out may be quarterly modified to monthly at deposit level, which now needs to be modified to quarterly. System will permit such modification. However monthly will not be permitted.  Throw message "Modification not permitted. Interest pay out frequency at product level is breached" in case modification is not permitted as mentioned above  Interest pay out frequency can be greater than the minimum frequency maintained at Product level @param frequency { <a href="#">@link Frequency</a> } @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @return { <a href="#">@link Frequency</a> } InterestFrequency required to calculate interest payout and compounding dates	This method calculates the net rate by adding the udm passed from pricing and validates the same against product level minimum and maximum  Net Interest rates eligible for a product and then updates the same over the deposit. @param arrangementNumber @param eventRateVarianceDTOArray

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @return { <a href="#">@link Frequency</a> } InterestFrequency required to calculate interest payout and compounding dates	The method returns the Interest Compounding Frequency which is present at interest definition rule. @param productId @return { <a href="#">@link Frequency</a> }
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @return { <a href="#">@link Frequency</a> } InterestFrequency required to calculate interest payout and compounding dates	This method fetches payout and compounding frequency details. @param contractDetailComputationRequest { <a href="#">@link ContractDetailComputationRequest</a> } Object containing parameters required to fetch frequency details @return { <a href="#">@link DepositFrequencyDetailsDTO</a> } Object containing deposit frequency details
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interestFrequencies need to calculate nextInterestpayoutdate and nextinterestCompoundingdate @param interestRuleKeyDTO { <a href="#">@link InterestRuleKeyDTO</a> } Object containing interest rule key parameters which identifies interest rule uniquely @return { <a href="#">@link Frequency</a> } InterestFrequency required to calculate interest payout	The method returns the Interest Compounding Frequency which is present at interest definition rule. @param productId @return { <a href="#">@link Frequency</a> }

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			and compounding dates	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCompoundingFrequency	This method retrieves the interest frequencies need to calculate nextInterestPayoutDate and nextInterestCompoundingDate @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link Frequency} InterestFrequency required to calculate interest payout and compounding dates	This method performs calculation of maturity amount and interest for the remaining balance after part redemption. @param remainingPrincipalBalance {@link Money} the principal balance after part redemption @param remainingTerm {@link Term} remaining term after redemption @param redemptionDate {@link Date} date on which redemption to be performed @param maturityDate {@link Date} date on which deposit will mature @param interestRate {@link BigDecimal} net interest rate applicable @param interestPayoutFrequency {@link Frequency} interest payout frequency of term deposit @param interestCompoundingFrequency {@link Frequency} interest compounding frequency of TD @param deposit {@link Deposit} Object containing term deposit details @return {@link InvestmentAvenue} Object containing amount and term details after simulation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	updatePayoutFrequency	This method updates the payoutFrequency and/or the maturity date of an already registered interest plan @param accountId Term deposit account id @param depositNumber Term deposit id @param interestClass {@link InterestClass} Higher level classification of interest rule @param interestDivision {@link InterestDivision} Lower level classification of interest rule @param effectiveDate Date used to fetch interest item plan @param maturityDate Maturity date of term deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency of TD @returns Date interest capitalization date	This method updates the payoutFrequency and/or the maturity date of an already registered interest plan @param accountId Term deposit account id @param depositNumber Term deposit id @param interestClass {@link InterestClass} Higher level classification of interest rule @param interestDivision {@link InterestDivision} Lower level classification of interest rule @param effectiveDate Date used to fetch interest item plan @param nextCapitalizationScheduledDate Next Capitalization Scheduled Date of the deposit. It will be maturity date if @param interestPayoutFrequency {@link Frequency} Interest payout frequency of TD @returns Date interest capitalization date @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#updatePayoutFrequency(java.lang.String, int, com.ofss.fc.enumeration.interest.InterestClass, com.ofss.fc.enumeration.interest.InterestDivision, com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date, com.ofss.fc.enumeration.td.Frequency)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	updatePayoutFrequency	This method updates the payoutFrequency and/or the maturity date of an already registered interest plan @param accountId Term deposit account id @param depositNumber Term deposit id @param interestClass {@link InterestClass} Higher level classification of interest rule @param interestDivision {@link InterestDivision} Lower level classification of interest rule @param effectiveDate Date used to fetch interest item plan @param maturityDate Maturity date of term deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency of TD @returns Date interest capitalization date	This method is used to update the deposit contract.  The customer requests for modification of Deposit amount and /or Deposit term of   term deposit within the window period (14 days default, defined at Brand and product level)  after Deposit placement or roll-over. The interest pay-out frequency can also be modified subject to Product rules  In contract modification allowed changes are  <li>Change in principal amount</li>  <li>Change in pay-out frequency</li>   <li>change in term</li> @param deposit {@link Deposit} Deposit details @param contractChangeHistory {@link ContractChangeHistory} Object containing history of contract modification @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing details of rate variance at particular event @param pricingTicket The pricing ticket @param transactionReferenceNumber the unique number for a given transaction. @param transactionSubsequenceNumber the unique number for a given transaction in pair with transactionReferenceNumber,  if particular transaction composed of two or more transactions or else transactionSubsequenceNumber will always be 1 @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updatePayoutFrequency	This method updates the payoutFrequency and/or the maturity date of an already registered interest plan @param accountId Term deposit account id @param depositNumber Term deposit id @param interestClass {@link InterestClass} Higher level classification of interest rule @param interestDivision {@link InterestDivision} Lower level classification of interest rule @param effectiveDate Date used to fetch interest item plan @param maturityDate Maturity date of term deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency of TD @returns Date interest capitalization date	This method updates the PayoutFrequency In Interest Plan if there is change in the frequency. @param deposit {@link Deposit} Term deposit details after contract modification @param existDeposit {@link Deposit} Deposit before contract modification
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestDetailsForInvestment	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest.	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest. @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#getInterestDetailsForInvestment(com.ofss.fc.app.td.dto.simulation.InvestmentSimulationRequestDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interest.	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestDetailsForInvestment	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest.	This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestDetailsForInvestment	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest.	This method calculates total interest earned and maturity amount @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing amount and term details after simulation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestDetailsForInvestment	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest.	This method performs calculation of maturity amount and interest for the remaining balance after part redemption. @param remainingPrincipalBalance {@link Money} the principal balance after part redemption @param remainingTerm {@link Term} remaining term after redemption @param redemptionDate {@link Date} date on which redemption to be performed @param maturityDate {@link Date} date on which deposit will mature @param interestRate {@link BigDecimal} net interest rate applicable @param interestPayoutFrequency {@link Frequency} interest payout frequency of term deposit @param interestCompoundingFrequency {@link Frequency} interest compounding frequency of TD @param deposit {@link Deposit} Object containing term deposit details @return {@link InvestmentAvenue} Object containing amount and term details after simulation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getInterestInquiryDetailsBeforeMaturityProcessing	This method calls method from Interest inquiry service to inquire details of interest for term deposit which includes unrecovered penalty amount. @param arrangementNumber Containing account and deposit id @return {@link MaturityProcessingInterestInquiryDTO} Object containing interest inquiry details	This method calls method from Interest inquiry service to inquire details of interest for term deposit which includes unrecovered penalty amount. @param arrangementNumber Containing account and deposit id @return {@link MaturityProcessingInterestInquiryDTO} Object containing interest inquiry details
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateBranchCode	There should be a facility to transfer TD account from one branch to another, if customer initiates the transfer. The account number will not change but the branch to which the account reports	There should be a facility to transfer TD account from one branch to another, if customer initiates the transfer. The account number will not change but the branch to which the account reports will change.  The changed branch id would be stored in the interest plan item.  The accounting entries for



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>will change. &lt;br&gt; The changed branch id would be stored in the interest plan item.&lt;br&gt; The accounting entries for capitalization, compounding or accrual either of which are pending would be passed into the new branch.&lt;br&gt; &lt;b&gt;Note :&lt;/b&gt; The accounting entries for the new branch would be in credit tag and the old branch it would be in debit tag.&lt;br&gt; The date for raise accounting event would be branchTransferDate minus 1 day Algorithm : &lt;ol&gt; &lt;li&gt;For the given arrangement number fetch all the interest plan item. &lt;li&gt;Loop through all the interest pan items that are fetched. &lt;ol&gt; &lt;li&gt;If the interest capitalizedAmount is non zero &lt;br&gt; Then raiseAccountingEvent for capitalization &lt;li&gt;If the interest CompoundedtillDateis non zero &lt;br&gt; Then raiseAccountingEvent for compounding &lt;li&gt;If the interest accrued till date is non zero &lt;br&gt; Then raiseAccountingEvent for accrual &lt;li&gt;Update the new branch id in the interest plan item. &lt;/ol&gt; &lt;/ol&gt; @param arrangementNumber @param transferBranchCode @param branchTransferDate @throws FatalException</p>	<p>capitalization, compounding or accrual either of which are pending would be passed into the new branch.&lt;br&gt; &lt;b&gt;Note :&lt;/b&gt; The accounting entries for the new branch would be in credit tag and the old branch it would be in debit tag.&lt;br&gt; The date for raise accounting event would be branchTransferDate minus 1 day Algorithm : &lt;ol&gt; &lt;li&gt;For the given arrangement number fetch all the interest plan item. &lt;li&gt;Loop through all the interest pan items that are fetched. &lt;ol&gt; &lt;li&gt;If the interest capitalizedAmount is non zero &lt;br&gt; Then raiseAccountingEvent for capitalization &lt;li&gt;If the interest CompoundedtillDateis non zero &lt;br&gt; Then raiseAccountingEvent for compounding &lt;li&gt;If the interest accrued till date is non zero &lt;br&gt; Then raiseAccountingEvent for accrual &lt;li&gt;Update the new branch id in the interest plan item. &lt;/ol&gt; &lt;/ol&gt; @param arrangementNumber @param transferBranchCode @param branchTransferDate @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateBranchCode	<p>There should be a facility to transfer TD account from one branch to another, if customer initiates the transfer. The account number will not change but the branch to which the account reports will change. &lt;br&gt; The changed branch id would be stored in the interest plan item.&lt;br&gt; The accounting entries for capitalization, compounding or accrual either of which are pending would be passed into the new branch.&lt;br&gt; &lt;b&gt;Note:&lt;/b&gt; The accounting entries for the new branch would be in credit tag and the old branch it would be in debit tag.&lt;br&gt; The date for raise accounting event would be branchTransferDate minus 1 day</p> <p>Algorithm :</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;For the given arrangement number fetch all the interest plan item. &lt;/li&gt;</li> <li>&lt;li&gt;Loop through all the interest pan items that are fetched. &lt;/li&gt;</li> <li>&lt;li&gt;If the interest capitalizedAmount is non zero &lt;br&gt; Then raiseAccountingEvent for capitalization &lt;/li&gt;</li> <li>&lt;li&gt;If the interest CompoundedtillDateis non zero &lt;br&gt; Then raiseAccountingEvent for compounding &lt;/li&gt;</li> <li>&lt;li&gt;If the interest accrued till date is non zero &lt;br&gt; Then raiseAccountingEvent for accrual &lt;/li&gt;</li> <li>&lt;li&gt;Update the new branch id in the interest plan item. &lt;/li&gt;</li> </ul> </li> </ul>	<p>This method updates the branch for the account and all of its regular deposits. This method when called via branch process. Then the deposit array and the termDepositAccount Entities are null. @param accountId @param newBranchId @param branchTransferDate @param Deposit [] {@link Deposit} @param termDepositAccount {@link TermDepositAccount} The method first validates the branch transferDate, The branch transfer date should be Greater than or equal to posting date and then it checks if the account is eligible for branch transfer&lt;br&gt; if the account is eligible for the transfer then the execution of the branch transfer happens by following the below steps &lt;ul&gt; &lt;li&gt;The list of deposits for the account is fetched by calling {@link DepositService} : fetchAllDeposits() method&lt;/li&gt; &lt;li&gt;Iterate through the deposit list and pick those deposits that has deposit status as regular&lt;/li&gt; &lt;ol&gt; &lt;li&gt;update the branchId of the deposit by calling updateDepositBranch() method&lt;/li&gt; &lt;li&gt;raise an Accounting Event by calling {@link TDAccountingAdapter} : raiseAccountingEvent&lt;/li&gt; &lt;li&gt;intimate the branch id change to Interest by calling {@link TDInterestAdapter} : updateDepositBranchInInterestPlanItem&lt;/li&gt; &lt;li&gt;intimate the branch change to tax by calling {@link TDTaxationAdapter} : updateDepositBrachInTax() &lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;finally after looping through the deposit list and updating the deposits&lt;br&gt; raise an alert by logging a nonfinancial Event by calling {@link TDEventLogService} :logNonFinancialEvent()&lt;/li&gt; &lt;li&gt;update the branch for the account by calling the updateAccountBranch&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> &lt;/ol&gt; @param arrangementNumber @param transferBranchCode @param branchTransferDate @throws FatalException </pre>	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCurrentCapitalizationStatistics		<p>This method will be used to inquire about the current capitalization statistics. This method will provide the current anniversary day and the current capitalization date. &lt;br&gt; The anniversary day would be returned only if it is an anniversary product otherwise the anniversary day would be returned as 0.&lt;br&gt; Please refer to {@link FrequencyBasis} for more information. &lt;br&gt; The response includes : &lt;ol&gt; &lt;li&gt;Current anniversary Day : Day on which the capitalization will take place(Applicable only in case of anniversary product) &lt;li&gt;Current capitalization processing date details : Please refer to {@link ProcessingDateDetails} &lt;/ol&gt; Algorithm:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Prepare InterestRuleKey using InterestRuleKeyDTO &lt;li&gt;Fetch {@link InterestRule} corresponding to {@link InterestRuleKey} using {@link InterestRuleResolutionService#fetchInterestRule()} method. &lt;li&gt;For given arrangement number, {@link InterestClass}, {@link InterestDivision}, fetch {@link InterestPlanItem} which is latest with respect to current posting date. &lt;li&gt;Create blank instance of {@link ProcessingDateDetailsDTO} &lt;li&gt;Create blank instance of {@link InterestCapitalizationStaticsResponse} &lt;li&gt;Fetch capitalization processingDateDetails from {@link CapitalizationStatistics#getNextCapitalizationDateDetails()} from the interest plan item. &lt;li&gt;Fetch accountLevelVariance from {@link InterestPlanItem#getAccountLevelVariance()} to get the anniversary day &lt;li&gt;Populate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ProcessingDateDetailsDTO using processingDateDetails &lt;li&gt;Set the {@link InterestCapitalizationStatisticsResponse#setProcessingDateDetailsDTO} == processingDateDetailsDTO &lt;li&gt;Set the {@link InterestCapitalizationStatisticsResponse#setCurrentAnniversaryDay} == accountLevelVariance.AnniversaryDay &lt;li&gt;Return the interestCapitalizationStatisticsResponse &lt;ol&gt; @param accountId @param depositId</p>
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCurrentCapitalizationStatistics		This method fetches the Interest related details of deposit @param depositAttributeInquiryDTO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getCurrentCapitalizationStatistics		This method updates the interest attributes like the anniversary date. @param deposit @param depositAttributeModificationDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getRevisedCapitalizationStatistics		<p>/ This method will be used by to inquire the revised capitalization statistics after account level modification is performed.&lt;br&gt; In certain cases the consumer may wish to modify the date or frequency on which interest is credited to consumer's account. &lt;br&gt; The frequency can only be increased by the consumer . The consumer may also decide that they want the changes to be applicable from the current cycle or from the next cycle.&lt;br&gt; In case the consumer wish to apply the changes from the current cycle then the revised Capitalization dates can be inquired through this method.&lt;br&gt;This method will not effect any data.</p> <p>&lt;br&gt; &lt;b&gt;Note &lt;/b&gt;: &lt;ol&gt; &lt;li&gt;If the consumer only want to change the anniversary day then eventFrequencyDTO send to this method should be null. &lt;li&gt;If the consumer only want to change the Frequency then anniversary day send to this method should be 0. &lt;/ol&gt; &lt;br&gt; Alorithm : &lt;li&gt;Prepare InterestRuleKey using InterestRuleKeyDTO &lt;li&gt;Fetch {@link InterestRule} corresponding to {@link InterestRuleKey} using {@link InterestRuleResolutionService#fetchInterestRule()} method. &lt;li&gt;For given arrangement number, {@link InterestClass}, {@link InterestDivision}, fetch {@link InterestPlanItem} using {@link InterestPlanItemService.fetchInterestPlanItemForGivenDate} which is latest with respect to current posting date. &lt;li&gt;Call {@link InterestOperationApplicationService#validateCapitalizationParametersModification(InterestRule,InterestPlanItem EventFrequencyDTO,anniversaryDay ,isApplicableInCurrentCycle,postingDate)} to validate the frequency modification. &lt;li&gt;Create blank instance of {@link ProcessingDateDetailsDTO} &lt;li&gt;Create blank instance of {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				InterestCapitalizationStaticsResponse} <li>&lt;li&gt;processingDateDetails = {@link InterestOperationApplicationService#deriveNextPhaseCapitalizationDetails(InterestRule, InterestPlanItem, EventFrequencyDTO, anniversaryDay, isApplicableInCurrentCycle)} &lt;/li&gt;Populate {@link ProcessingDateDetailsDTO} using processingDateDetails &lt;/li&gt;Set the {@link InterestCapitalizationStatisticsResponse#setmodifiedCapitalizationProcessingDateDetailsDTO} =processingDateDetailsDTO &lt;/li&gt;Set the {@link InterestCapitalizationStatisticsResponse#setmodifiedAnniversaryDay} = anniversaryDay from input &lt;/li&gt;Return the interestCapitalizationStatisticsResponse &lt;/ol&gt;  @param accountId @param depositId @param anniversaryDay @param capitalizationCycle @param interestDivision @return</li>
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getRevisedCapitalizationStatistics		The method fetches the revised new Capitalized Dates @param depositAttributeInquiryDTO @return @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	getRevisedCapitalizationStatistics		This method updates the interest attributes like the anniversary date. @param deposit @param depositAttributeModificationDTO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	modifyCapitalizationStatistics		This method updates the new anniversary date details for interest capitalisation. @param accountId @param depositId @param anniversaryDay @param capitalizationCycle @param interestDivision

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	modifyCapitalizationStatistics		This method updates the interest attributes like the anniversary date. @param deposit @param depositAttributeModificationDTO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getAccrualFrequency		This method retrieves the Accrual Frequency for given Interest Rule key @param interestRuleKeyDTO {@link InterestRuleKeyDTO} Object containing interest rule key parameters which identifies interest rule uniquely @return {@link Frequency} InterestFrequency required to calculate interest payout and compounding dates @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#getAccrualFrequency(com.ofss.fc.app.interest.dto.interestrule.InterestRuleKeyDTO)
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getAccrualFrequency		This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	logInterestActionForTDPMI		This method logs the interest action in the interest eod actions table for interest division post maturity. @param accountId @param depositId @param pickupDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	logInterestActionForTDPMI		<p>This method marks the deposit as mature. The Balances are updated according to the Principal Payout Option. if <a href="#">PrincipalPayoutOption#NO_INSTRUCTION</a> is set then :   <ol style="list-style-type: none"> <li>1. Principal Balances are moved to Uncollected Principal Balance. <a href="#">Balance#uncollectedPrincipalBalance</a> += <a href="#">Balance#principleBalance</a> + <a href="#">Balance#BalancePrincipalDue</a></li> <li>Interest Balances are moved to Uncollected Interest. <a href="#">Balance#uncollectedInterestBalance</a> += <a href="#">Balance#BalanceInterestDue</a></li> </ol> The following buckets are zeroised : <a href="#">Balance#principleBalance</a>, <a href="#">Balance#BalancePrincipalDue</a>, <a href="#">Balance#BalanceInterestDue</a></p> <p>If <a href="#">PrincipalPayoutOption#RE_DEEM</a> is set then :   <ol style="list-style-type: none"> <li>1. Principal Balances are moved to Principal Balance Due. <a href="#">Balance#balancePrincipalDue</a> += <a href="#">Balance#principleBalance</a> + <a href="#">Balance#uncollectedPrincipalBalance</a></li> <li>Interest Balances are moved to Interest Balance Due. <a href="#">Balance#balanceInterestDue</a> += <a href="#">Balance#uncollectedInterestBalance</a></li> </ol> The following buckets are zeroised : <a href="#">Balance#principleBalance</a>, <a href="#">Balance#uncollectedPrincipalBalance</a>, <a href="#">Balance#uncollectedInterestBalance</a> </p> <p>@param deposit</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	getNumberOfDaysForTaxCalculation	This method calculates number of days for which tax needs to be calculated. @param accountId @param depositId @param interestDivision @param redemptionDate @param maturityDate @param taxApplicableAt @return	This method inquires redemption details and will also calculate the maturity outstanding penalty amount, tax applicable and net interest payable on remaining deposit after redemption. @param redemptionRequestDTO {@link RedemptionRequestDTO} Object containing details required for premature redeem TD simulation @return {@link PrematureRedeemTDDTO} Object containing redemption details and maturity, interest details in case of premature part redeem simulation
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateDepositStatus	This method is used to update the account status in interest plan for given arrangement number @param accountId @param depositId	Based on the flag isRedemptionRenewalforUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePricipalDue and UncollectedPricipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDInterestAdapter	updateDepositStatus	This method is used to update the account status in interest plan for given arrangement number @param accountId @param depositId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	isCompoundingApplicable	This method will determine whether compounding is applicable to the arrangement/product specified. The input is either the product code or the account number and deposit number. The method will fetch the IRD corresponding to the input and check whether it contains the compounding frequency. @param accountId @param depositNumber @param effectiveDate TODO @return @throws FatalException	
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	isCompoundingApplicable	This method will determine whether compounding is applicable to the arrangement/product specified. The input is either the product code or the account number and deposit number. The method will fetch the IRD corresponding to the input and check whether it contains the compounding frequency. @param accountId @param depositNumber @param effectiveDate TODO @return @throws FatalException	This method uses adapter for interest capitalization and also logs TDFinancialItemStatement. @param actionRequest @param accountId @param depositId @param dueDate @param maturityDate TODO @throws FatalException
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	fetchTaxComputationDates		This method fetches all the TermDeposit events and the processing date on which the events will happen @param casaAccountId @param fromDate @param toDate @return
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	validateDepositOpeningDetails	This method is used to validate the conditions required to open a backdated Term Deposit. It checks if the first tax computation date is backdated. If so then the deposit creation should not be allowed.The	This method checks if the Funding is skipping a Tax Occurrence Date.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>inputs to this method are:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;productId&lt;/li&gt;</li> <li>&lt;li&gt;branchId&lt;/li&gt;</li> <li>&lt;li&gt;interestDivisionDTO: The interestDivisionDTO should have effective date as a mandatory input. The payout frequency and capitalization due date may be sent by the module during maturity.&lt;/li&gt;&lt;br&gt; The steps are as follows: &lt;br&gt; &lt;ol&gt; &lt;li&gt;The productId for which the deposit needs to be created comes as an input. This is used to fetch the interest rules linked to the particular productId. &lt;li&gt;The interest rule with {@link InterestDivision#REGULAR} is the one that is going to be used in the deposit opening. &lt;li&gt;Then we derive the interestProcessingStartDate from the {@link InterestDivisionDTO#getEffectiveDate()} coming in as input and the interest rule derived by calling the method {@link InterestPlanItemService#getInterestProcessingStartDate(Date, InterestRule)}. &lt;li&gt;Now we derive the {@link EventFrequencyDTO} for capitalization by calling the method {@link InterestPlanRegistrationService#constructCapitalizationFrequencyDTO(I</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>interestDivisionDTO, InterestRule)) which takes the interestDivisionDTO and interestRule as input. &lt;li&gt;Make a call to the {@link InterestPlanRegistrationService#validateFirstComputationTaxDateForTD()} to check the conditions required to open a backdated TD Deposit are met with. &lt;li&gt;Deposit opening wont be allowed if any of the following conditions are true:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If {@link TaxApplicableAt#ACCRUAL} and the first accrual should have already happened in which case Tax would already be computed. &lt;li&gt;If {@link TaxApplicableAt#COMPOUNDING} and the first compounding should have already happened in which case Tax would already be computed. &lt;li&gt;If {@link TaxApplicableAt#CAPITALIZATION} and the first capitalization should have already happened in which case Tax would already be computed.&lt;/li&gt; &lt;/ol&gt; &lt;/ol&gt;</li> </ul> <p>@param productId @param branchId @param effectiveDate @param interestPayoutFrequency {@link Frequency} @param MaturityDate</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDInterestAdapter	validateDepositOpeningDetails	<p>This method is used to validate the conditions required to open a backdated Term Deposit. It checks if the first tax computation date is backdated. If so then the deposit creation should not be allowed. The inputs to this method are:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;productId&lt;/li&gt;</li> <li>&lt;li&gt;branchId&lt;/li&gt;</li> <li>&lt;li&gt;interestDivisionDTO: The interestDivisionDTO should have effective date as a mandatory input. The payout frequency and capitalization due date may be sent by the module during maturity.&lt;/li&gt;</li> </ul> <p>The steps are as follows:</p> <ol style="list-style-type: none"> <li>&lt;li&gt;The productId for which the deposit needs to be created comes as an input. This is used to fetch the interest rules linked to the particular productId. &lt;li&gt;The interest rule with {@link InterestDivision#REGULAR} is the one that is going to be used in the deposit opening. &lt;li&gt;Then we derive the interestProcessingStartDate from the {@link InterestDivisionDTO#getEffectiveDate()} coming in as input and the interest rule derived by calling the method {@link InterestPlanItemService#getInterestProcessingStartDate(Date, </li></ol>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>InterestRule)). &lt;li&gt;Now we derive the {@link EventFrequencyDTO} for capitalization by calling the method {@link InterestPlanRegistrationService#constructCapitalizationFrequencyDTO(InterestDivisionDTO, InterestRule)} which takes the interestDivisionDTO and interestRule as input. &lt;li&gt;Make a call to the {@link InterestPlanRegistrationService#validateFirstComputationTaxDateForTD()} to check the conditions required to open a backdated TD Deposit are met with. &lt;li&gt;Deposit opening wont be allowed if any of the following conditions are true:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If {@link TaxApplicableAt#ACCRUAL} and the first accrual should have already happened in which case Tax would already be computed. &lt;li&gt;If {@link TaxApplicableAt#COMPOUNDING} and the first compounding should have already happened in which case Tax would already be computed. &lt;li&gt;If {@link TaxApplicableAt#CAPITALIZATION} and the first capitalization should have already happened in which case Tax would already be computed.&lt;/li&gt;</li> </ul> <p>@param productId @param branchId @param effectiveDate</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param interestPayoutFrequency {@link Frequency} @param MaturityDate	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	@param repaymentStage @param accountParameters @param redrawDate @param rescheduleOption @param interestRates @param allRepaymentStages @param retainRepaymentDay @param interestBaseHistory @param scheduleTransactionHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method gets StageEnd Date On TermReschedule @param repaymentStage @return Date
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This function will call DB package PkLnEventSimulation to fetch simulation schedule definition and details @param account @param effectiveDate @param simulationType @param penaltyMethodType @param PPFAmount @param expectedPayOffAmount @param rescheduleOptionType @param rateVarSlab @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method returns the result on adding the given period into given start date. It returns appropriate date based on the passed year base type @param startDate @param period @param yearBase @param dayOfMonthToPreserve @return Date lastOccurrenceDate
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	@param repaymentSchedule @param changedStageNumber @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	@param userDefinitionDate @param changedStageDefinition @param accountParameters @param interestRates @param changedStageParameter @param interestBaseHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method redefines the stage when reschedule is done @param repaymentStage @param accountParameters @param redrawDate @param rescheduleOption @param interestRates @param allRepaymentStages @param retainRepaymentDay @param interestBaseHistory @param scheduleTransactionHistory



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method redefines the stage when reschedule is done by recalculating the installment. @param repaymentStage @param interestRates @param allRepaymentStages @param presentValue @param redrawDate @param retainRepaymentDay @param interestBaseHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method gets StageEnd Date On TermReschedule @param repaymentStage @return Date
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method validates the stage if one of its parameter is changed. On successful validation, it also resets other stage attributes which are affected due to change in the given stage parameter @param changedStage @param accountParameters @param onDate @param futureValue @param interestRates @param parameterType @param interestBaseHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	Defines the given repayment stage.  This method internally calculate & set :-  1. first principal repayment date   2. first interest repayment date   3. installment amount   4. no. of principal repayments   5. no. of interest repayments   6. Stage end date   7. first interest capitalisation date  8. first fee capitalisation date  @param repaymentStage
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method gets StageEnd Date On TermReschedule @param repaymentStage @return Date
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method redefines the stage when reschedule is done by recalculating the installment. @param accountParameters @param repaymentStage @param interestRates @param allRepaymentStages @param presentValue @param redrawDate @param retainRepaymentDay @param interestBaseHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getLoanTransactionAttribute	This method will get the value of loan transaction attribute from the session. @param loanAttributeType @return	This method validates the stage if one of its parameter is changed. On successful validation, it also resets other stage attributes which are affected due to change in the given stage parameter @param changedStage @param accountParameters @param onDate @param futureValue @param interestRates @param parameterType @param interestBaseHistory

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	addLoanTransactionAttribute	This method will add the loan transaction attribute and its value in the session. @param loanAttributeType @param value	@param accountNo @param effectiveDate @param simulationType @param penaltyMethodType @param PPFAmount @param expectedPayOffAmount @param rescheduleOptionType @param udmFixed @param udmVariable @param sessionId @param applicationContext @return @throws FatalException
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	addLoanTransactionAttribute	This method will add the loan transaction attribute and its value in the session. @param loanAttributeType @param value	
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		@param repaymentStageToBeGenerated @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getAccountIdForScheduleId		This method generates repayment records for the given stage   In EIPI Schedule, interest & fee has not been charged to user but capitalised at capitalisation dates to outstanding principal amount.  @param repaymentStage @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, LoanRepaymentRecordDTO> repaymentRecords

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		@param repaymentStageToBeGenerated @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This function will takes argument for setting inquiry response for Schedule inquiry @param account @param definition @param details @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This is the common method to perform reschedule on rate change, subsequent disbursement and partial payoff. @param oldRepaymentStages @param accountParameters @param rescheduleDate @param splitDate @param rescheduleOption @param interestRates @param flgCapitalizeArrears @param interestBaseHistory @param scheduleTransactionHistory @return LoanRescheduleOutputData reschedule output data

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method returns the new schedule on account deferment or deferment by interest freeze @param oldRepaymentStages @param accountParameters @param defermentDate @param splitDate @param isDefermentByInterestFreeze @param interestFreezeRateValue @param installmentRuleCodeForDeferment @param installmentTypeForDeferment @param stageDescription @param defermentTerm @param endOfPeriodTreatment @param interestRates @param interestBaseHistory @param scheduleTransactionHistory @return PrincipalScheduleDefermentOutputData deferment output
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		@param allRepaymentStages @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method logs schedule transactions for interest base @param oldRepaymentStages @param accountParameters @param rescheduleDate @param currentStage @param flgCapitalizeArrears @param amountForCapitalization @param interestBaseHistory @param scheduleTransactionHistory

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method updates the schedule if one of the repayment stages is changed  Method is used to apply the effects of changing one of the allowed parameters of a particular stage to the next repayment stages in order. @param repaymentSchedule Repayment stages @param changedStageNumber Number of stage of which one of the parameter is changed @param accountParameters Parameters from loan account, account balances, required for schedule generation @param onDate Date of stage parameter change @param interestRates Interest rate plans @param interestBaseHistory List of interest base records @param scheduleTransactionHistory List of scheduled transactions @return LoanRescheduleOutputData rescheduleOutput
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method calculates the interest amount. This is used during schedule generation. @param accountParameters @param interestBaseHistory @param interestCalculationDateRange @param interestRates @param loanRepaymentStage @return Money @throws FatalException
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method add rollover interest, if any, from the previous stage to total principal amount to be repaid for the given stage @param repaymentStage @param accountParameters @param schedule @param interestBaseHistory @param scheduleTransactionHistory
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method generates repayment records for the given stage @param repaymentStageToBeGenerated @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, RepaymentRecord>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				repaymentRecords
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		@param accountParameters @param amountForScheduleGeneration @param onDate @param scheduleTypeCode @param repaymentDay @param newDisbursedAmount @param debitTypeFeeAmount @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		Common method for blown up schedule generation for new schedule and on reschedule/restructure/deferment/subsequent disbursement/top up disbursement @param sessionContext @param scheduleId @param sessionId @param rescheduleDate @param rescheduleOption @return LoanBlownupScheduleResponse @throws FatalException
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method saves blown up schedule generation output to temporal tables @param sessionId @param blownupOutput
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		Method for blown up schedule generation for new schedule and on reschedule/restructure/deferment/subsequent disbursement/top up disbursement @param accountParameters @param generationDate @param newRepaymentSchedule @param startingPrincipalBalance @param interestRates @param rescheduleOption @param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				scheduleTransactionHistory @param interestBaseHistory @return LoanAccountScheduleData
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		@param disbursementData @param amountForScheduleGeneration @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		<p>&lt;p&gt;This method is responsible for defining the Stages for Schedule Generation. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Divides given amount for the schedule generation equally across all the principal recovery stages in the given schedule type.&lt;/li&gt; &lt;li&gt;Divides the loan term, from given account schedule parameters, across stages based on term validation rule from the schedule type&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; Following are the major actions taken by this method:- &lt;br&gt; &lt;ol&gt; &lt;li&gt;It accepts Arrangement Id&lt;/li&gt; &lt;li&gt;If the the provided arrangement id is NULL, then it is being derived from Loan Account Additional Details&lt;/li&gt; &lt;li&gt;The schedule configuration is searched for the arrangement id. &lt;br&gt; This schedule configuration has been defined as per user requirement.&lt;/li&gt; &lt;li&gt;If there is no custom schedule configuration defined, then the default configuration is being referred for defining Stages.&lt;/li&gt; &lt;li&gt;If the custom schedule configuration has been defined already, then it's being referred for defining the stages.&lt;/li&gt; &lt;/ol&gt; &lt;/p&gt;           @param accountParameters @param amountForScheduleGeneration @param onDate @param scheduleTypeCode @param arrangementId @return         </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method generates the blown up Schedule when reschedule is done. It preserves the elapsed blown up schedule records based on the reschedule option passed in the argument. @param rescheduleDate @param accountParameters @param newRepaymentSchedule @param totalPrincipalToRepay @param interestRates @param rescheduleOption @param oldBlownupSchedule @param scheduleTransactionHistory @param interestBaseHistory @return LoanAccountScheduleData blownupScheduleOutput
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method does reschedule for the given repayment schedule. This method doesn't save the new schedule @param oldRepaymentSchedule Indicates existing schedule for the loan account @param accountParameters Loan account specific parameters required for schedule generation @param rescheduleDate Indicates date of reschedule @param rescheduleOption Indicates reschedule option type. Recalculate Term / Installment. @param interestRates Interest rate plans to be used for reschedule @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @param accountEventType Indicates loan account event type @return LoanRescheduleOutputData rescheduleOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		<p>This method will be called at the time backdated reschedule/ restructure event as well as replay events. It takes inputs like account id, event date, event based history number &amp; new schedule code (in case of backdated restructure). This method calls - A] PrincipalSchedulePeriodHistoryRepository for schedule configuration at the time of reschedule either based on event based history number or event date. B] LoanBalanceAggregateRepository to fetch schedule principal balance at the time reschedule or replay event. C] LoanScheduleTypeRepository to fetch schedule type details in case of backdated restructure where new schedule code is supplied by the framework. D] doBackdatedReschedule() method to generate summary schedule E] generateBackdatedRecordsOnReschedule() method to generate blown up schedule. F] saveBackdatedSchedule to save the backdated schedule in the main table. @param accountId @param eventDate @param eventNumber @param newScheduleCode</p>
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		<p>This method is used to save the backdated schedule in the main table. Also this method will insert (at the time of backdated reschedule / restructure) or update (at the time of replay events) history tables. @param accountId @param sessionId @param onDate @param rescheduleOption</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method is used to generate summary schedule. This method calls LoanBackdatedRescheduleService for - A] defineBackdatedStages which uses the provided schedule configuration information like number of stages, installment rule code, year base type, frequencies of stages etc. to generate summary schedule. B] defineBackdatedSchedule which takes care of moratorium interest & installment amount. C] doBackdatedReschedule which actually reschedules the account based on reschedule option type. (For backdated reschedule, reschedule option type will always be RECALCULATE_INSTALLMENT) @param accountId @param rescheduleDate @param schedulePrincipalBalance @param rescheduleOption @param accountEventType @param capitalizeArrears @param loanRepaymentScheduleConfiguration @param accountParameters
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method is used to generate blown up schedule. This method calls LoanBackdatedRescheduleService for - A] generateBackdatedRecords which uses the backdated schedule generated by doBackdatedReschedule method to generate backdated blown up schedule. @param accountId @param sessionId @param repaymentStages @param scheduleInterestBase @param scheduleTransaction @param rescheduleDate @param rescheduleOption @param schedulePrincipalBalance @param accountEventLogType @param accountParameters
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method constructs LoanRepaymentScheduleConfigurationDTO from the supplied loan repayment stage records. @param loanRepaymentStages

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		@param repaymentStageDTO @param accountParameters @param interestRatesForStage @return
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method calculates the interest amount. This is used during schedule generation. @param accountParameters @param interestBaseHistory @param interestCalculationDateRange @param interestRates @param loanRepaymentStage @return Money @throws FatalException
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method generates repayment records for the given stage   In EIPI Schedule, interest & fee has not been charged to user but capitalised at capitalisation dates to outstanding principal amount.  @param repaymentStage @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, LoanRepaymentRecordDTO> repaymentRecords

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleAccountAdapter	getScheduleId		This method calculates the number of repayments based on the given stage term, installment amount, loan present value, interest rate and repayment frequency. @param repaymentStage @param presentValue @param interestRates @param actionDate @return
IScannedLogEmailAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\logscanner\IScannedLogEmailAdapter	sendMail		
IScannedLogEmailAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\logscanner\IScannedLogEmailAdapter	sendMail		
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	validateLoanAccountAndFetchDetails	This method fetches the status of a loan account. @param accountId @return	This method validates the alternate credit account. <ol> If account type is DDA <ol> Check if required data is present in the map .If not fetch the details and validate </ol> <li>Existence of account</li> <li>Product Group Nature (Should be asset or liability)</li> <li>System defined status of account (Only regular and inactive are valid status)</li> <li>User defined status of account (Should not have any user defined status)</li> </ol> </ol> <ol> If account type is Loan, validate the following <li>Existence of account</li> <li>Status of the account.( Should be regular)</li> </ol> In case of any validation error, appropriate validation error is added to the list of validation errors. The details of the account are added to the map and the map is returned. @param parentAccountId @param accountIdForCreditPosting @param accountType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param map @return HashMap
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	checkOffsetAccountLinkage	This method fetches the array of Loan Accounts for whom, the input account acts as Offset Account. @param accountId @return	This method fetches the array of Loan Accounts for whom, the input account acts as Offset Account. @param accountId @return
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	checkOffsetAccountLinkage	This method fetches the array of Loan Accounts for whom, the input account acts as Offset Account. @param accountId @return	This method will be used inquire the linkage details of the account with another CASA, TD or Loan account. The method works as follows : <ul style="list-style-type: none"> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForFees} to fetch the list of parent accounts for which the CASA acts as an alternate account for fees posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForDebitInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for debit interest posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForCreditInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for credit interest posting.</li> <li>Call the method {@link IDDALoanAdapter#checkOffsetAccountLinkage} to get the list of Loan accounts for which the CASA acts as an Offset account.</li> <li>Call the method {@link IDDALoanAdapter#checkLoanAccountPayInstruction} to get the list of Loan accounts for which the CASA acts as an installment payment account.</li> <li>Call the method {@link } to get the list of credit card numbers for which the CASA acts as a payment account.</li> <li>Call the method {@link IDDASettlementAdapter#fetchTDPayoutLinkageDetailsForDDAccount} to get the list of Term Deposit accounts and Deposit numbers for which the CASA</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				acts as payout account for Principal or Interest component.</li> <li>Combine the response received by each of the methods in a single collection along with the linkage type and return.</li> </ul> @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	checkLoanAccountPayInstruction	This method fetches the array of Loan Accounts for whom, the input account acts as Payment Account. @param accountId @return	This method fetches the array of Loan Accounts for whom, the input account acts as Payment Account. @param accountId @return
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	checkLoanAccountPayInstruction	This method fetches the array of Loan Accounts for whom, the input account acts as Payment Account. @param accountId @return	This method will be used inquire the linkage details of the account with another CASA, TD or Loan account.The method works as follows : <ul> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForFees} to fetch the list of parent accounts for which the CASA acts as an alternate account for fees posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForDebitInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for debit interest posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForCreditInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for credit interest posting.</li> <li>Call the method {@link IDDALoanAdapter#checkOffsetAccountLinkage} to get the list of Loan accounts for which the CASA acts as an Offset account.</li> <li>Call the method {@link IDDALoanAdapter#checkLoanAccountPayInstruction} to get the list of Loan accounts for which the CASA acts as an installment payment account.</li> <li>Call the method {@link } to get the list of credit card numbers for which the CASA acts as a payment

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>account.</p> <ul style="list-style-type: none"> <li>Call the method <a href="#">{@link IDDASettlementAdapter#fetchTDPayoutLinkageDetailsForDDAccount }</a> to get the list of Term Deposit accounts and Deposit numbers for which the CASA acts as payout account for Principal or Interest component.</li> <li>Combine the response received by each of the methods in a single collection along with the linkage type and return.</li> </ul> <p>@param accountId</p>
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	creditFunds	Method to credit funds in loan account. @param accountId @param amountToCredit @param populateAccountingEventAmountDebit	<p>This method is used to credit funds to a demand deposit account identified by accountTransaction.accountId. This method works as follows:</p> <ol style="list-style-type: none"> <li>Loads the demand deposit account by calling <a href="#">{@link DemandDepositAccountRepository#load}</a> using accountTransaction.accountId.</li> <li>transactionAmounts = <a href="#">{@link TransactionAmounts#prepareTransactionAmounts()}</a></li> <li>Call <a href="#">{@link DemandDepositAccount#credit()}</a>.</li> <li>Update the instance of DemandDepositAccount by calling the update() method of the repository <a href="#">{@link DemandDepositAccountRepository#update}</a></li> </ol> <p>//Construct and log the financial statement item StatementItemService.addDDFinancialStatementItem()</p> <p>To work around the running balance bug specified in <a href="#">{@link DemandDepositAccount#finalEventBasedBalanceSnapshots}</a>, we take a snapshot of the account balances before credit begins and simply add the credit amount to the clear balance of this map.This is stored in <a href="#">{@link DemandDepositAccount#setBalancesForCredit()}</a>. This can be later on used to prepare the EarmarkRecovery map during apportionment at <a href="#">{@link</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FundsApportionmentService} or to prepare the sweepin map during acquisition at {<a href="#">@link FundsAcquisitionService</a>}. &lt;/br&gt; Once the actual credit is done, we check for the following two things:</p> <ol style="list-style-type: none"> <li>If the FinancialEventType = SweepIn. In this case we adjust the balance snapshot that was placed in the {<a href="#">@link DemandDepositAccount#finatialEventBasedBalanceSn apshots</a>} against SweepIn, with the credit response. Refer {<a href="#">@link #adjustSnapshotAfterSweepIn()</a>}.</li> <li>Else if the credit response has a non-zero earmark amount apportioned, then we dont clear the {<a href="#">@link DemandDepositAccount#balancesForCredit</a>} bucket. It will be cleared in rest all the cases. &lt;/ol&gt; So when {<a href="#">@link DemandDepositAccount#getAccountBalancesForFincialEventType()</a>} is called to log balance buckets in statement, we get the corresponding snapshot for the current transaction. The balacesForCredit bucket is then cleared after logging statement. &lt;/p&gt; @<a href="#">@param accountTransaction</a> {<a href="#">@link AccountTransaction</a>} This attribute holds basic details required for crediting funds into the account.&lt;br&gt; These are the mandatory values : &lt;ul&gt; <li>{<a href="#">@link AccountTransaction#getAccountId()</a>}</li> <li>{<a href="#">@link AccountTransaction#getFinancialEventType()</a>}</li> <li>{<a href="#">@link AccountTransaction#getServiceId()</a>}</li> <li>{<a href="#">@link AccountTransaction#getTransactionAmounts()</a>}</li> <li>{<a href="#">@link AccountTransaction#getTransactionReferenceNumber()</a>}</li> <li>{<a href="#">@link AccountTransaction#getTransactionSubsequenceNumber()</a>}</li> <li>{<a href="#">@link AccountTransaction#getTransactionSubsequenceNum</a></li> </li></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> ber())&lt;/li&gt; &lt;li&gt;{@link AccountTransaction#getTransactionAttributes()}&lt;/li&gt; &lt;ul&gt; &lt;li&gt;{@link TransactionAttributes#getMessageType()}&lt;/li&gt; &lt;li&gt;{@link TransactionAttributes#getTransactionType()}&lt;/li&gt; &lt;/ul&gt; &lt;li&gt;{@link AccountTransaction#}&lt;/li&gt; &lt;/ul&gt; @return {@link CreditResponse} which contains information about the funds apportionment. This method can may throw these {@link BusinessException} if the provided data is wrong. &lt;ul&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCO UNT_TRANSACTION_NULL}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_TRANS ACTION_ATTRIBUTES_NULL}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_AMOU NT_NOT_PRESENT}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_NEGAT IVE_TRANSACTION_AMOUNT}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_INVALI D_TRANSACTION_TYPE}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_INVALI D_MESSAGE_TYPE}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_INVALI D_DATES}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_INVALI D_ACCOUNT_STATUS}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ERROR _IN_FUNDS_APPORTIONMENT_STRATEGY}&lt;/li&gt; &lt;/ul&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	creditFunds	Method to credit funds in loan account. @param accountId @param amountToCredit @param populateAccountingEventAmountDebit	This method is used to transfer funds from one demand deposit account (provider account) to another demand deposit account (beneficiary account).   A funds transfer is always initiated from the provider account.  A funds transfer results in the following:  Dr Provider Account Cr Beneficiary Account The amount to be transferred can be expressed either in terms of the provider account currency or the beneficiary account currency.  <ol> <li>Amount expressed in provider account currency: In this case, the amount to be transferred to the beneficiary account is derived, in case it is not sent by the caller. <li> Amount expressed in beneficiary account currency: In this case, the amount to be transferred from the provider account is derived, in case it is not sent by the caller. </li> </ol> This method works as follows:  <ol> <li>Construct AccountTransaction object for provider account. accountAmount = providerAmount. transactionAmount = beneficiaryAmount. Change transaction description to include beneficiary account number. Call this.debitFunds().</li> <li>Construct AccountTransaction object for beneficiary account. accountAmount = beneficiaryAmount. transactionAmount = providerAmount. Change transaction description to include provider account number. Call this.creditFunds().</li> </ol> @param fundTransferTransaction @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	creditFunds	Method to credit funds in loan account. @param accountId @param amountToCredit @param populateAccountingEventAmountDebit	This method is used to credit funds to a demand deposit account identified by accountTransaction.accountId.  The method differs from the current service creditFunds only with respect to the financial statement item. It will not log a financial statement item, since by default the statement item is logged on cheque deposit date. This method works as follows: <ol> <li>Loads the demand deposit account by calling {@link DemandDepositAccountRepository#load} using accountTransaction.accountId.</li> <li>transactionAmounts = {@link TransactionAmounts#prepareTransactionAmounts()}</li> <li>Call {@link DemandDepositAccount#credit()}.</li> <li>Update the instance of DemandDepositAccount by calling the update() method of the repository {@link DemandDepositAccountRepository#update}</li></ol> @param accountTransaction
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	creditFunds	Method to credit funds in loan account. @param accountId @param amountToCredit @param populateAccountingEventAmountDebit	This method is used to credit funds on credit interest settlement. This function specially created because we do not want to call {@link #creditFunds} method as it does not take value date < process date. But this case is possible in case of settlement.   Consider following situation -  Credit Settlement date: 31-Jan-2011,  31-Jan, 1-Feb, 2-Feb-2011 falls on holiday. So this settlement will be processed in the BOD of 3-Feb-2011.  In this case Transaction.PostingDate = 3-Feb-2011  Transaction.valueDate=1-Feb-2011   So in this case valuedate < processDate. , {@link #creditFunds} method does not allow this case.  This method is created to overcome this condition. As we do not want to consider this transaction as backdated transaction. This method will perform following activity-   1)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>TransactionDates settlementTransactionDates = new TransactionDates()  transactionDates.postingDate =  input.accountTransaction.TransactionDates.postingDate  transactionDates.valueDate =  input.accountTransaction.TransactionDates.ValueDate  transactionDates..transactionDate =  input.accountTransaction.TransactionDates.transactionDate.  2) TransactionDates  creditTransactionDates = new TransactionDates()  transactionDates.postingDate =  input.accountTransaction.TransactionDates.postingDate  transactionDates.valueDate =  input.accountTransaction.TransactionDates.postingDate  transactionDates..transactionDate  =input.accountTransaction.TransactionDates.transactionDate.  3) AccountTransaction.transactionDates = creditTransactionDates.. As creditFunds gives error on backdates we change the date  4) load DemandDepositAccount-  5) Call demandDepositAccount.creditFunds method to credit the funds.  6) AccountTransaction.transactionDates = settlementTransactionDates.  7) prepareDDFinancialStatementItem(AccountTransaction)-- In statement we want value date as settlement date + 1. @param accountTransaction</p>
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDALoanAdapter	creditFunds	Method to credit funds in loan account. @param accountId @param amountToCredit @param populateAccountingEventAmountDebit	Intimate Loan on earmark recovery. The method must call Loan's credit service. @param earmark @param populateAccountingEventAmountDebit @param amountToCredit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDALoanAdapter	intimateOnOffsetBalanceChange	The method will be called to intimate the loan account which has been marked as beneficiary for a CASA offset account on balance change in the offset account. @param accountId	This method will be used to levy period fees on the account on the fees charging date as intimated by e-pricing. This service will levy the fees on the main account or the alternate account if one has been maintained for fees posting.   It is assumed that when the service is called, the periodic fees to be charged on the input.chargingDate has already been computed and stored in e-pricing. Hence, it calls the service to retrieve the computed price and subsequently levies the fees on the account.   <pre> It works as follows : <ul><li> Call the service {@link IDDA PricingAdapter#getComputedPeriodicFees} to get the calculated periodic fees. </li><li> Call the service {@FeesManagementService#levyPeriodicFees} with the periodic fee details returned by the above service to levy the fees on the account. </ul></pre> @param applicationContext @param accountId @param chargingDate



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDALoanAdapter	intimateOnOffsetBalanceChange	The method will be called to intimate the loan account which has been marked as beneficiary for a CASA offset account on balance change in the offset account. @param accountId	Refer the class level documentation for this. <ul> <li>This method will first call the method {@link IDemandDepositAccount#checkAndMoveBalances}  <li>While updating the current book balances it needs to check whether the Future Book Balances for the same date are moved into the current book balances.  <li>The offset clear and unclear balance would be sent to interest for computation <li>If Future Book Balances for the next date also exist at that time, the current book balances will be updated with those Future Book Balances if the effective dates of both the balances are same.  <li>After updating the current book balances, balances history will be maintained, thus an instance of the balance w.r.t the effective date. <li>If the account is not OD,the interest computation process is invoked to calculate interest on the current balances.  <li>For OD accounts, this process will be done in the proportionate OD shell. Refer {@link IntimateInterestActionExecutor}</li>.<li>This method need not to bother about the interest computation parameters, it will simply call {@link InterestFacade#invokeInterestComputation()}.<li>While computing interest, interest will always be computed on current balances.  <li>Refresh the balance statistics by calling account.statistics.refreshForBalances In BalanceChangeActionExecutor, we will set the crossover field of BalanceInformationDTO either to true or false, depending upon the different amounts(TOD,OD,Overline,AUF). This method works as follows :   <pre> Load DemandDepositAccount = {@link DemandDepositAccountRepository#load()} Call {@link IDemandDepositAccount#checkAndMoveBalances()}</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> Balancechangeactionexecutor will use BalanceInformationDTO and will set the different types of Amount. New attribute isCredit is added in DemandDepositBalances, so that we can set the crossover field to true in case of crossover. if ((demandDepositAccount.getUtilizedAUFLimit(tranD ate).isPositive()    demandDepositAccount.getUtilizedOverdraftAmount (tranDate).isPositive()    demandDepositAccount.getOverlineAmount(tranDate ).isPositive()    demandDepositAccount.getTODAmount(tranDate).is Positive()) &amp;&amp;&amp; demandDepositAccount.getIsCredit()) { balanceInformationDTO.setIsCrossOver(true); demandDepositAccount.setIsCredit(false); } else if (demandDepositAccount.getUtilizedAUFLimit(tranDa te).isZero() &amp;&amp;&amp; demandDepositAccount.getUtilizedOverdraftAmount (tranDate).isZero() &amp;&amp;&amp; demandDepositAccount.getOverlineAmount(tranDate ).isZero() &amp;&amp;&amp; demandDepositAccount.getTODAmount(tranDate).is Zero() &amp;&amp;&amp; !demandDepositAccount.getIsCredit()) { balanceInformationDTO.setIsCrossOver(true); demandDepositAccount.setIsCredit(true); } Prepare BalanceHistoryItem by calling parameterised constructor of BalanceHistoryItem() Call {@link BalanceHistoryItemRepository#saveOrUpdate()} Call {@link InterestFacade#invokeInterestComputation()} Call {@link DemandDepositAccount#AccountStatistics#refreshFor Balances} &lt;/pre&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	intimateOnOffsetBalanceChange	The method will be called to intimate the loan account which has been marked as beneficiary for a CASA offset account on balance change in the offset account. @param accountId	The method will log actions on linked accounts if any for a balance change on the parent account after BalanceChange action. Logging BalanceChangeAction for current date is not an option at this point as it will never be picked up. As logging a BalanceChangeAction directly may result in unwanted cascade effect for linked accounts of the linked accounts, we avoid this logging more specific actions like offset adjustment or proportionate od actions. It does the following: <ol style="list-style-type: none"> <li>If the account is an offset account, log offset adjustment action on the beneficiary account for next BOD.</li> <li>If the account is an OD account, log proportionate OD action for all the linked accounts in the next BOD</li> </ol> @param iDemandDepositAccount @param effectiveDate @param pickupDate @param epochType
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	The method will be used for processing changes in the limit in eLCM. Whenever, any change in the facility takes places, it impacts the balance buckets at the account level. The change maybe : <ol style="list-style-type: none"> <li>Limit expiry</li> <li>Increase of limit amount.</li> <li>Decrease of limit amount.</li> <li>Change in the expiry date of the limit.</li> </ol> When any of the above changes takes place in the account, reapportionment of the balance buckets is required. The method works as follows: <ul style="list-style-type: none"> <li>Inquire the account linked to the input facilityId by calling the repository method {@link DemandDepositAccountRepository#fetchAccountByFacilityId}</li> <li>Load the account by calling the repository method {@link DemandDepositAccountRepository#load}</li> <li>Call the method {@link DemandDepositAccount#processBalanceMovement} to reapportion the balance buckets.</li> </ul> @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				referenceNumber -This parameter is the unique identifier for each transaction. @param facilityId - This parameter is used to identify Overdraft Facility. </ul>
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method is used when there is OD Linkage Before Expiry. This method will change the od expiry date in demand deposit account. @param accountId @param accountFacilityLinkage
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for TD principal payout where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for TD maturity where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Loan disbursement where the account number passed is beneficiary account number in Loan disbursement setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Sweep Out where the account number passed is beneficiary account number/provider account number in Sweep Out setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method will be used to trigger all the validations for account closure. The closure is not permitted if these conditions are present in the account. The method works as follows : <ul style="list-style-type: none"> <li>Inquire the linkages of the account by calling the service {@link this.fetchAccountLinkages}</li> <li>Inquire the status of the facilities of the account by calling the service {@link this.fetchActiveAccountFacilities}</li> <li>Inquire the balances of the account by calling the service {@link this.fetchAccountBalances}</li> <li>Inquire the alternate accounts for the account by calling the service {@link AlternateAccountApplicationService#fetchAlternateAccountDefinitionForAnAccount}</li> <li>Call the service {@link this.calculateSettlementAmount} which will calculate and return the final balance in the account which has to be settled.</li> <li>Set all the above inquired values in the DDAClosureBusinessPolicyDTO</li> <li>Create the instance of the business policy class by calling the service {@link DDAClosureBusinessPolicyFactory#createPolicyInstance}</li> <li>Call the method {@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				DDAClosureBusinessPolicy#validate} to perform the validations.</li> </ul> @param accountId @param isAccountBeingClosed @param pricingMap @throws FatalException
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	<p>TODO: The methods for credit card payment account and TD payout account are pending. This method will be used inquire the linkages of the account with another CASA, TD or Loan account. The CASA account may be linked to another account in the following ways :</p> <ul style="list-style-type: none"> <li>The account may act as an alternate account for another account for its fee and interest posting.</li> <li>It may act as an offset account for a loan account. In this case, the interest is calculated on the (principal amount - balance in the offset account). So the interest calculated is reduced in such a case.</li> <li>It may be linked to a loan account for the payment of installments.</li> <li>It may act as a payment account for credit cards of the account holder.</li> <li>It may act as a payout account for the principal and interest component of a Term Deposit.</li> </ul> <p>The method works as follows :</p> <ul style="list-style-type: none"> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForFees} to check if the account acts as an alternate account for another CASA for its fee component. If the size of array returned &gt; 0, then populate the map accountLinkagesMap&lt;Fees,True&gt; else accountLinkagesMap&lt;Fees,False&gt;</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForDebitInterest} to check if the account acts as an alternate account for another CASA for its debit interest posting. If the size of array returned &gt; 0, then populate the map</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>accountLinkagesMap&lt;DebitInterest,True&gt; else  accountLinkagesMap&lt;DebitInterest,False&gt;&lt;/li&gt;  &lt;li&gt;Call the method {@link  AlternateAccountService#fetchParentAccountsForCreditInterest} to check if the account acts as an alternate  account for another CASA for its credit interest  posting. If the size of array returned &gt; 0, then  populate the map  accountLinkagesMap&lt;DebitInterest,True&gt; else  accountLinkagesMap&lt;CreditInterest,False&gt;&lt;/li&gt;  &lt;li&gt;Call the method {@link  IDDALoanAdapter#checkOffsetAccountLinkage} to  check if the account acts as an offset account for  another loans account. If the size of array returned &gt; 0,  then populate the map  accountLinkagesMap&lt;LoanOffset,True&gt; else  accountLinkagesMap&lt;LoanOffset,False&gt;&lt;/li&gt;  &lt;li&gt;Call the method {@link  IDDALoanAdapter#checkLoanAccountPaymentInstruction} to check if the account acts as a payment account  for another loans account for its installments. If the  size of array returned &gt; 0, then populate the map  accountLinkagesMap&lt;LoanInstallment,True&gt; else  accountLinkagesMap&lt;LoanInstallment,False&gt;&lt;/li&gt;  &lt;li&gt;Call the method {@link } to check if the account  acts a payment account for the credit cards of the  account holder. If the size of array returned &gt; 0, then  populate the map  accountLinkagesMap&lt;CreditCard,True&gt; else  accountLinkagesMap&lt;CreditCard,False&gt;&lt;/li&gt;  &lt;li&gt;Call the method {@link  IDDASettlementAdapter#fetchTDPayoutLinkageDetailsForDDAccount } to check if the account is a payout  account for Term Deposit for its Principal or Interest  component.If the size of array returned &gt; 0, then</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>populate the map  accountLinkagesMap&lt;TDPayout,True&gt; else  accountLinkagesMap&lt;TDPayout,False&gt;&lt;/li&gt; &lt;/ul&gt;  @param accountId</p>
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	<p>This method will provide the from offer code for given account details  @param accountId @return ITermDepositAccount</p>	<p>This method will be used inquire the linkage details of the account with another CASA, TD or Loan account.The method works as follows : &lt;ul&gt; &lt;li&gt;Call the method {@link AlternateAccountService#fetchParentAccountsForFees} to fetch the list of parent accounts for which the CASA acts as an alternate account for fees posting.&lt;/li&gt; &lt;li&gt;Call the method {@link AlternateAccountService#fetchParentAccountsForDebitInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for debit interest posting.&lt;/li&gt; &lt;li&gt;Call the method {@link AlternateAccountService#fetchParentAccountsForCreditInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for credit interest posting.&lt;/li&gt; &lt;li&gt;Call the method {@link IDDALoanAdapter#checkOffsetAccountLinkage} to get the list of Loan accounts for which the CASA acts as an Offset account.&lt;/li&gt; &lt;li&gt;Call the method {@link IDDALoanAdapter#checkLoanAccountPayInstruction} to get the list of Loan accounts for which the CASA acts as an installment payment account.&lt;/li&gt; &lt;li&gt;Call the method {@link } to get the list of credit card numbers for which the CASA acts as a payment account.&lt;/li&gt; &lt;li&gt;Call the method {@link IDDASettlementAdapter#fetchTDPayoutLinkageDetailsForDDAccount} to get the list of Term Deposit accounts and Deposit numbers for which the CASA acts as payout account for Principal or Interest component.&lt;/li&gt; &lt;li&gt;Combine the response received</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				by each of the methods in a single collection along with the linkage type and return.</li> </ul> @param accountId
IDDALoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDALoanAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method will be used to get the different components of account balance as well as the interest, fees and tax to be posted to the account.   By summing up all these balances, the net amount to be paid to the customer will be arrived during closure. The method works as follows : <ul> <li>Get the details of the DemandDepositAccount by calling the repository method {@link DemandDepositAccountRepository#load}.</li> <li>Get the clear balance of the account by calling the getter method {@link DemandDepositAccount#getClearBalance()} and populate it in the balance map.</li> <li>Get the AUF amount by calling the method {@link DemandDepositAccount#getUtilitizedAUFLimit} and populate it in the balance map.</li> <li>Get the hold amount by calling the method {@link DemandDepositAccount#getTotalHoldAmount} and populate it in the balance map.</li> <li>Get the unclear funds by calling {@link DemandDepositAccount#getCurrentUnclearFunds} and populate it in the balance map</li> <li>Get the utilizedOverdraft amount by calling {@link DemandDepositAccount#getUtilitizedOverdraftAmount} and populate it in the balance map</li> <li>Get the TOD amount by calling {@link DemandDepositAccount#getTODAmount} and populate it in the balance map</li> <li>Get the Overline amount by calling {@link DemandDepositAccount#getOverlineAmount} and populate it in the balance map</li> <li>Get the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>SweepIn lien amount by calling {@link DemandDepositAccount#getTDSweepInLienAmount} and populate it in the balance map&lt;/li&gt; &lt;li&gt;Call the method {@link IDDAInterestAdapter#inquireInterestAtArrangement Closure} with the interest class as Credit to get the credit interest due for the account. The method will also return the tax applicable on interest. Populate the details in the balanceMap.&lt;/li&gt; &lt;li&gt;Call the method {@link IDDAInterestAdapter#inquireInterestAtArrangement Closure} with the interest class as Debit to get the debit interest due for the account. Populate the amount in the balanceMap.&lt;/li&gt; &lt;li&gt;Inquire the alternate accounts for the account by calling the service {@link AlternateAccountApplicationService#fetchAlternateAccountDefinitionForAnAccount}&lt;/li&gt; &lt;li&gt;Call the service {@link this.calculateSettlementAmountForAccountClosure} to derive the closing balance of the account without considering the fees for the account.&lt;/li&gt; &lt;li&gt;Call the service {@link DemandDepositAccount#refreshAccountClosureAverageBalances} to refresh the average balance of the account. If the value of the input parameter isAccountBeingClosed = True, then send the value of the parameter inquiry as False, else send the value of inquiry as True to the service. The service in turn returns a parameter AverageBalance of the account.&lt;/li&gt; &lt;li&gt;Compare the parameter averageBalance returned by the above service with {@link DemandDepositAccount#getMinimumAverageBalance}. If the averageBalance &lt; {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DemandDepositAccount#getMinimumAverageBalance), then include the parameter in the call to pricing given below.</p> <ul style="list-style-type: none"> <li>Check if the input.pricingMap has a pricingTicketNumber for the Periodic fees.</li> <li>If the value of pricingTicketNumber is null, then call the service <code>IDDA PricingAdapter#.getPriceForAccountClosure</code> to get the total periodic fees due from the account. Else if the value of the input parameter is not null, it implies that the periodic fees for the account has been calculated. Hence, call the service <code>IDDA PricingAdapter#getFeesWithTicketNumber</code> to get the details of the periodic fees.</li> <li>Open a loop for DTO returned by the service and add the fees for each price feature to arrive at the total fees due from the account. Populate the amount in the balanceMap.</li> <li>Check if the input.pricingMap has a pricingTicketNumber for the Event fees.</li> <li>If the value of pricingTicketNumber is not null, call the service <code>IDDA PricingAdapter#getFeesWithTicketNumber</code> to get the details of the fee for the event account closure. Add the returned fee amount to the total fees due from the account and update in the balanceMap.</li> <li>Return the balanceMap.</li> </ul> <p>@param accountId @param isAccountBeingClosed @param pricingMap @throws FatalException</p>
ICommonServiceDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceDocumentationAdapter	fetchDocumentType		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\commons\service\ICommonServiceDocumentationAdapter	fetchDocumentType		This represents the fetchVerificationType method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,VerificationTypeDTO,verificationTypeDTO, ,verificationTypeDTO TODO: Explain the verificationTypeDTO above @return (@link com.ofss.fc.app.common\service\core\dto\verification.VerificationTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDomainCategoryTemplateProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IDomainCategoryTemplateProcessAdapter	startProcess	This method initiates the DomainCategoryTemplateProcess @param domainCategory String	
IDomainCategoryTemplateProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IDomainCategoryTemplateProcessAdapter	startProcess	This method initiates the DomainCategoryTemplateProcess @param domainCategory String	
IRuleExecutionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IRuleExecutionAdapter	executeRules		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IRuleExecutionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IRuleExecutionAdapter	executeRules		
IRuleExecutionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IRuleExecutionAdapter	executeRules		
IPMULoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IPMULoansAdapter	isRepaymentScheduleCodeValid	@param allowedRepaymentTypes @param allowedRepaymentScheduleCode @param productCode @param interestRuleDTOList @return	
IPMULoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IPMULoansAdapter	isRepaymentScheduleCodeValid	@param allowedRepaymentTypes @param allowedRepaymentScheduleCode @param productCode @param interestRuleDTOList @return	@param loanOffer
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResources	This method will load all the policy resources @throws FatalException	This method will execute DelayedTransactionPoller @throws FatalException
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResources	This method will load all the policy resources @throws FatalException	This method will execute load-resources for channels @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResources	This method will load all the policy resources @throws FatalException	
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResources	This method will load all the policy resources @throws FatalException	Initialize all security related singletons @param sessionContext @throws FatalException
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResourcesForChannels	This method will load all the policy resources for channel @throws FatalException	This method will execute load-resources for channels @throws FatalException
IAccessibleResourceLoaderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessibleResourceLoaderAdapter	loadResourcesForChannels	This method will load all the policy resources for channel @throws FatalException	
ILimitsAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	fetchBrokerForFacilities	This methods will map the corresponding AgentArrangementLinkage DTO to the passed facilityID @param appContext @param facilityId @param agentArrangementType @return @throws FatalException	AgentArrangementLinkageDTO
ILimitsAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	addAgentArrangementLinkage	This method will add the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO	This method will add the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return @throws FatalException	
ILimitsAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	addAgentArrangement Linkage	This method will add the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException	This method will update the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException
ILimitsAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	addAgentArrangement Linkage	This method will add the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException	This method is used to create a new facility in the Limits and Management system. The facility created can be of any of the types Credit (OD), Temporary Excess and Unclear Funds Facility. The type in the facility DTO identifies the type of facility being created. The add facility method also has the restrictions that can be created for the facility. Apart from the creation of the facility, system will also ensure that the facility effective limit is calculated and the facility value dated and book dated balances are also created. The Loan to Value ratio for the facility is also calculated along with the creation and stamped at the facility level. In case fees are linked to the facility, system interacts with the Pricing system to create the pricing plan. Users can stamp the brokers/introducers for the facilities and this information captured along with creation task and will be updated in the Agent module. In case of temporary excess or the uncleared facility LCM notifies CASA about the creation of the facility for the account and also the details pertaining to the facility like the amount, expiry date etc @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FacilityDTO,facilityDTO, ,facilityDTO. This DTO {@link FacilityDTO} contains the details for the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				creation. @return {@link TransactionStatus} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	updateAgentArrangementLinkage	This method will update the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException	This method is used to update the existing facility in the system.This method is applicable for all the different types of facilities. The update of the facility will recalculate the effective limit and the book dated and value dated balances of the facility and also the loan to value ratio. In case of any amount related change, expiry date change it notifies the OD regarding the same if the account linked to the facility is OD. Also Collections update is done in case of any updates to the collateral. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FacilityDTO,facilityDTO, ,This DTO {@link FacilityDTO} contains the details for the update. @return {@link TransactionStatus} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	fetchAgentArrangementLinkage	This method will fetch the agent arrangement details for the credit facility @param sessionContext @param AgentArrangementLinkageDTO @return @throws FatalException	This represents the fetchByVersion method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FacilityDTO,facilityDTO, ,This DTO {@link FacilityDTO} contains the details for the fetch. @return {@link FacilityInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAgentAdapter	fetchUltimateFacilityId	This method will fetch the ultimate facility Id for the credit facility @param sessionContext @param creditFacilityId @return @throws FatalException	This represents the fetchByVersion method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FacilityDTO,facilityDTO, ,This DTO {@link FacilityDTO} contains the details for the fetch. @return {@link FacilityInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isCutOffRunning	This method checks if cut-off is running for bank code and branch code under consideration @return, boolean	This method checks if cut-off is running for bank code, branch code under consideration if its not batch mode.
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isCutOffRunning	This method checks if cut-off is running for bank code and branch code under consideration @return, boolean	This method will check if cut-off is running or not and return boolean value. @return
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isBatchRunning	This method checks if EOD is running for bank code and branch code under consideration. @return, boolean	This method checks if Batch is running or not for bank code, branch code under consideration if its not batch mode.
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isBatchRunning	This method checks if EOD is running for bank code and branch code under consideration. @return, boolean	This method is used to black-out transactions invoked from online channels during batch or cut-off. If the transaction is getting invoked in batch mode, it will not be blackout out. @param sessionContext @param blackoutDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isBatchRunning	This method checks if EOD is running for bank code and branch code under consideration. @return, boolean	This method will check if EOD is running or not and return boolean value. @return
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isLoggedInTellerBatchClosed	This method checks if teller batch is closed for logged in user or not. @return, boolean	This method checks if teller batch is closed for logged in user. Returns true, if its closed. Else false
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isLoggedInTellerBatchClosed	This method checks if teller batch is closed for logged in user or not. @return, boolean	
IBatchSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IBatchSecurityAdapter	isLoggedInTellerBatchClosed	This method checks if teller batch is closed for logged in user or not. @return, boolean	This method checks if batch is closed for the session context user or not. @param sessionContext @return
IPricingEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingEventProcessingAdapter	processAdhocAlert	send adhoc alerts @param adhocAlertDTO @throws FatalException	@see com.ofss.fc.app.adapter.pricing.IPricingEventProcessingAdapter#processAdhocAlert(com.ofss.fc.app.adapter.pc.dto.AdhocAlertDTO)
IPricingEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingEventProcessingAdapter	processAdhocAlert	send adhoc alerts @param adhocAlertDTO @throws FatalException	Dispatches adhoc pricing alerts @param pricingAlertFor @param message

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt; </pre> <p>@param accountingEventDTO @throws FatalException</p>	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseAndProcessAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &lt;br&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt; &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &lt;br&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt; &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	This method will transfer money to GL from CASA account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &lt;br&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt; &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	This method will transfer money to GL from CASA account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\broop\IBranchAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;. </pre> @param accountingEventDTO @throws FatalException	
ILoanOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanOriginationAdapter	inquireSettlementDetails	@param sessionContext @param applicationId @return SettlementDetailsInquiryResponse	
ILoanOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanOriginationAdapter	inquireSettlementDetails	@param sessionContext @param applicationId @return SettlementDetailsInquiryResponse	@param sessionContext @param applicationId @return
ILoanOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanOriginationAdapter	inquireSettlementDetails	@param sessionContext @param applicationId @return SettlementDetailsInquiryResponse	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICCIWelcomeLetterAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\ICCIWelcomeLetterAdapter	fetchCCIDetailsForApplication	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation. @param sessionContext @param applicationId @return @throws FatalException	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation.
ICCIWelcomeLetterAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\ICCIWelcomeLetterAdapter	inquireOverDraftSimulationForSimulationId	This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String simulationId, , Indicates the identifier of the Simulation for which the selected simulation details are being fetched @return {@link ODSimulationResponse} This class contains the information that is retrieved for the simulation from draft for the term deposit account that is to be passed on during	This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			account opening. @throws FatalException	
ICCIWelcomeLetterAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\ICCIWelcomeLetterAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionContext @param insuranceId @return @throws FatalException	This service will fetch CCI premium quote amount for an Insurance on basis of InsuranceID.
IJournalTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalTDAAdapter	fetchAccountAttributes	@param sessionContext @param accountId @return termDepositAccountInquiryResponse @throws FatalException	
IJournalTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalTDAAdapter	fetchAccountAttributes	@param sessionContext @param accountId @return termDepositAccountInquiryResponse @throws FatalException	
IJournalTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalTDAAdapter	fetchAccountAttributes	@param sessionContext @param accountId @return termDepositAccountInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param secondaryReason	<p>This method will be invoked when the Fraud assessor believes that the intended beneficiary is fraudulent. In such case, the transaction has to go through authorization. But the provider account must be debited, or earmarked for funds so that the effect of the outgoing payment is felt immediately from the provider user experience. </p> <p> Hence, the system will first earmark the account with the given amount and commit the transaction. </p> <p> Post authorization, the system will invoke the following The Fraud assessor will provide a reference number in the ThreadLocal aliased as FRAUD_ASSERTION_TRANSACTION_ID. This id will be used as earmark reference number. The same will be used to check against CASA to remove the earmark while doing the main processing. </p>
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param secondaryReason	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param context @param accountID @param earmarkAmount @return Indicate the hold reference no. @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			secondaryReason	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param secondaryReason	<p>This method will be invoked when the Fraud assessor believes that the intended beneficiary is fraudulent. In such case, the transaction has to go through authorization. But the provider account must be debited, or earmarked for funds so that the effect of the outgoing payment is felt immediately from the provider user experience. </p> <p> Hence, the system will first earmark the account with the given amount and commit the transaction. </p> <p> Post authorization, the system will invoke the following The Fraud assessor will provide a reference number in the ThreadLocal aliased as FRAUD_ASSERTION_TRANSACTION_ID. This id will be used as earmark reference number. The same will be used to check against CASA to remove the earmark while doing the main processing. </p>
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param secondaryReason	This method will try to block funds on account. This will be applicable to CASA type of accounts only @param netAmount Indicates the amount that should be earmarked @param holdReferenceNo Indicates the reference number that should be used to track the hold request. @param earmarkParametersId TODO @param primaryReason TODO @param secondaryReason TODO @param accountId Indicates the CASA Account ID @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFunds	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log @param accountId @param amount return Indicate the hold reference no. @param earmarkParametersId @param primaryReason @param secondaryReason	This method will handle the outgoing payment transaction on network holiday. This method will check if the activation date for the transaction is a holiday in the network calendar, then without checking cut-off, set the transaction as executed post cut-off and raise held over accounting event. This will handle the post cut off transaction and place a hold on the funds of the remitter account, if the holds funds are enabled at the template level for the post cutoff transactions. When the transaction is picked up during the batch processing this will remove the hold on the remitter account for the hold reference number of the respective transaction. @param transactionContext @param transactionLog @param transactionTemplate @param messageLogItem @param fundsExchangeDateDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param pcAccountTransactionDTO @param nsftTreatmentType	This method will perform credit to the CASA account for which business payment was initiated, in case an error is encountered during procurement of funds or perform debit from the CASA account for which business collection was initiated, in case an error is encountered while debiting an account which belongs to the same bank.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, boolean, isFirstCall <li>Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.</li> @return AccountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param pcAccountTransactionDTO @param nsftTreatmentType	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param pcAccountTransactionDTO @param nsftTreatmentType	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param pcAccountTransactionDTO @param nsftTreatmentType	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	<p>This method will update the clear balance of the provider account for Business Collection transactions.</p> @fcb.param MI, ApplicationContext, applicationContext, Standard header information for the application. @fcb.param MI, ITransactionLog, transactionLog, Indicates the log being processed. @fcb.param MI, MessageLogItem, messageLogItem, Indicates the message being processed.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	This method will initiate outgoing collection from the GL, which was debited as part of the credit accounting processing, for each debit entry in the instruction list.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, BulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.BulkTransferInstruction}</li> @throws BusinessException @throws PaymentsBatchException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	clearFunds	This method will move the uncleared funds to clear balance. @param pcAccountTransactionDTO @throws FatalException	This method will remove any the uncleared funds to the extent of provided amount. The amount will be same as the beneficiary amount from the transaction log. It should also generate a entry in account audit trail as a negative aspect of the updateUnclearFunds @param accountID @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeDeposit	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will perform credit operation on the CASA account for which business payment transaction has been initiated.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @return {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeDeposit	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeDeposit	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeDeposit	This method will add the provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeReversal	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will perform credit to the CASA account for which business payment was initiated, in case an error is encountered during procurement of funds or perform debit from the CASA account for which business collection was initiated, in case an error is encountered while debiting an account which belongs to the same bank.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, boolean, isFirstCall <li>Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.</li> @return AccountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	updateBalanceOnChequeReversal	This method will reverse adding of provided amount to the unclear balance. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmarkAndDebit	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo @param moduleReferenceNo
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmarkAndDebit	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This method will undo the effect of any earlier done transaction, which has now undergone dishonor or reject. This method should not be used for reversal sense, which is usually done on the same day on the original transaction @param transactionLog @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmarkAndDebit	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param accountTransactionDTO @throws FatalException	This will try to remove the earmark on the CASA account using the hold reference number, that used initially to create the earmark @param transactionContext @param log @param isDebitRequired @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeUnclearedFunds	This method will remove any the uncleared funds to the extent of provided amount. The amount will be same as the beneficiary amount from the transaction log. It should also generate a entry in account audit trail as a negative aspect of the updateUnclearFunds @param accountTransactionDTO @throws FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeUnclearedFunds	This method will remove any the uncleared funds to the extent of provided amount. The amount will be same as the beneficiary amount from the transaction log. It should also generate a entry in account audit trail as a negative aspect of the updateUnclearFunds @param accountTransactionDTO @throws FatalException	This method will remove any the uncleared funds to the extent of provided amount. The amount will be same as the beneficiary amount from the transaction log. It should also generate a entry in account audit trail as a negative aspect of the updateUnclearFunds @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	debitFunds	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param accountTransactionDTO @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	debitFunds	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param accountTransactionDTO @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @throws FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	debitFunds	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param accountTransactionDTO @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @throws FatalException	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @param context @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	debitFunds	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param accountTransactionDTO @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @throws FatalException	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo @param moduleReferenceNo

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	debitFunds	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param accountTransactionDTO @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the acocunt under force post conditions @throws FatalException	This method will perform debit operation on the CASA account for which business payment transaction has been initiated.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @return {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	reverseDebitFunds	This method will reverse the debit action on the account. It should also generate a entry in account audit trail @param accountTransactionDTO	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	reverseDebitFunds	This method will reverse the debit action on the account. It should also generate a entry in account audit trail @param accountTransactionDTO	This method will reverse the debit action on the account. It should also generate a entry in account audit trail @param context @param accountID @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	reverseCreditFunds	This method will reverse the credit action on the account. It should also generate a entry in account audit trail @param accountTransactionDTO	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	reverseCreditFunds	This method will reverse the credit action on the account. It should also generate a entry in account audit trail @param accountTransactionDTO	This method will be called when a credit transaction processed for CASA account is to be reversed. @param pcAccountTransactionDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	@param transactionParameters @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	This method will remove the hold and debit the account. The hold no is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo @param moduleReferenceNo
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	This method will remove hold from an account. The hold no. is provided in the moduleReferenceNo. The hold no is obtained when the earmarkFunds was done earlier. @param accountId @param earmarkReferenceNumber @return transactionStatus
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	@param transactionParameters @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	This method will undo the effect of any earlier done transaction, which has now undergone dishonor or reject. This method should not be used for reversal sense, which is usually done on the same day on the original transaction @param transactionLog @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	This will try to remove the earmark on the CASA account using the hold reference number, that used initially to create the earmark @param transactionContext @param log @param isDebitRequired @return @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	@param sessionContext @param accountId @param holdReferenceNo @return transactionStatus @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	removeEarmark	This method will remove earmark from an already earmarked account. @param accountId @param earmarkReferenceNumber @return @throws FatalException	This method will handle the outgoing payment transaction on network holiday. This method will check if the activation date for the transaction is a holiday in the network calendar, then without checking cut-off, set the transaction as executed post cut-off and raise held over accounting event. This will handle the post cut off transaction and place a hold on the funds of the remitter account, if the holds funds are enabled at the template level for the post cutoff transactions. When the transaction is picked up during the batch processing this will remove the hold on the remitter account for the hold reference number of the respective transaction. @param transactionContext @param transactionLog @param transactionTemplate @param messageLogItem @param fundsExchangeDateDTO @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	notifyBackdatedRepricing	This method is used as trigger point whenever back dated pricing change is done @param accountId @param effectiveDate @return @throws FatalException	This method will call the demand deposit notification adapter {@link IPaymentsNotificationAdapter} and log the Backdated Repricing action @param accountNo @param effectiveDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	notifyBackdatedRepricing	This method is used as trigger point whenever back dated pricing change is done @param accountId @param effectiveDate @return @throws FatalException	This method is used as trigger point whenever back dated pricing change is done @param accountId @param effectiveDate @return @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	transferFunds	This method will be called for book transfer transactions in case of outgoing payments. @param fundTransferDTO @return @throws FatalException	This method is used to transfer funds from the customer account by calling the respective CASA service @param accountID the account id to be used for transfer funds @param txnAmount the transaction amount to be used for transfer funds @param currencyCode the currency code to be used @param fundTransferDTO carry the accountID,txnAmount,currencyCode data to perform the @return TransactionStatus if transfer funds to CASA failed or was success
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	transferFunds	This method will be called for book transfer transactions in case of outgoing payments. @param fundTransferDTO @return @throws FatalException	This method will transfer funds with in own bank accounts @param fundTransferDTO @param transactionLog @param messageLogItem @param transactionContext @param messageIdentification @return @throws FatalException
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	logStatementForReturnTransactions	@param transactionDTO @throws FatalException	This method will log transaction into the CASA tables for return transactions. @param postingDate @param copiedTxnLog @param eventId @param messageLogItem @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFundsForOutgoingPayment	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log. This is executed when a outgoing payment transaction is performed post-cutoff. This places a hold for such transactions and intimate the activate date to the user. The hold is removed during the activation date batch processing. @param accountId @param earmarkAmount @param holdReferenceNumber @param earmarkParametersId @param primaryReason @param secondaryReason @return @throws FatalException	
IPaymentsCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsCASAAdapter	earmarkFundsForOutgoingPayment	This method will place hold on the clear balance to the extent of the provided amount. The amount will be same as the remitter amount from the transaction log. This is executed when a outgoing payment transaction is performed post-cutoff. This places a hold for such transactions and intimate the activate date to the user. The hold is removed during the activation date batch processing. @param accountId @param earmarkAmount @param holdReferenceNumber @param earmarkParametersId @param primaryReason @param secondaryReason @return @throws FatalException	This method will handle the outgoing payment transaction on network holiday. This method will check if the activation date for the transaction is a holiday in the network calendar, then without checking cut-off, set the transaction as executed post cut-off and raise held over accounting event. This will handle the post cut off transaction and place a hold on the funds of the remitter account, if the holds funds are enabled at the template level for the post cutoff transactions. When the transaction is picked up during the batch processing this will remove the hold on the remitter account for the hold reference number of the respective transaction. @param transactionContext @param transactionLog @param transactionTemplate @param messageLogItem @param fundsExchangeDateDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	fetchPartyA ssetClassific ation		This will fetch the PartyAssetClassificationInquiryResponse for party. @param partyAssetClassificationDTO @return
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	fetchPartyA ssetClassific ation		@param preliminaryRestrictionDTOs @return
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	fetchPartyA ssetClassific ation		This will return the applicability level of asset classification @return
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	fetchPartyA ssetClassific ation		This will check whether CRR code exists against the party.
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	getAllRestri ction		This will fetch the allowed list of services of particular CRR code. @param crrCode @return
IPartyNPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyNPAA dapter	checkAppli cabilityLeve l		This will check whether CRR code exists against the party.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDBreakCostRetrievalAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDBreakCostRetrievalAdapter	getEconomicCost	This method is used to fetch economic cost, charged if Term Deposit is redeemed before maturity date. @param economicCostInquiryRequestDTO {@link BreakCostInquiryRequest} Object containing details required for derivation of economic cost @return {@link BreakCostInquiryResponse} Object containing details of economic cost	@see com.ofss.fc.app.adapter.AdapterFactory#getAdapter(java.lang.String, com.ofss.fc.datatype.NameValuePair[])
ITDBreakCostRetrievalAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDBreakCostRetrievalAdapter	getEconomicCost	This method is used to fetch economic cost, charged if Term Deposit is redeemed before maturity date. @param economicCostInquiryRequestDTO {@link BreakCostInquiryRequest} Object containing details required for derivation of economic cost @return {@link BreakCostInquiryResponse} Object containing details of economic cost	This method is used to fetch economic cost, charged if Term Deposit is redeemed before maturity date. This method will return a default response with Zero economic amount. It should be implemented by subclasses of this adapter @param economicCostInquiryRequestDTO {@link BreakCostInquiryRequest} Object containing details required for derivation of economic cost @return {@link BreakCostInquiryResponse} Object containing details of economic cost @see com.ofss.fc.app.adapter.td.ITDBreakCostRetrievalAdapter#getEconomicCost(com.ofss.fc.app.adapter.td.dto.BreakCostInquiryRequest)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	fetchInterestRateDetails	<p>During Interest Computation interest rates used can be passed as input.&lt;br&gt; If these rates are not passed as input then Interest module derives the rates from pricing using the facts given in input {@link FactContext}. This method will derive the rates and return rate details inside {@link InterestDetailsDTO}. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. @param arrangementNumber Arrangement number which is registered in Interest module by the consuming module. @param eventId Identifier of event for which rates needs to be fetched. @param evaluationDate Date for which interest rates needs to be fetched. @param baseForInterest TODO</p>	<p>When Interest rates to be used for interest computation are not given as input, Interest module need to get them from Pricing module.&lt;br&gt;This method is used to fetch interest rate details from Pricing for given arrangement number and {@link InterestConstituentType}.&lt;br&gt; It calls {@link IInterestPricingAdapter} to derive interest rates from Pricing. {@link FactContext} and evaluation Date needs to be passed as input. @param factContext {@link FactContext} containing values of facts required for deriving rates from pricing. @param arrangementNumber Arrangement number which is registered in Interest module by the consuming module. @param interestConstituentType {@link InterestConstituentType} for which Interest Rate needs to be derived. @param evaluationDate Date with respect to which Interest rate needs to be derived. return {@link InterestDetailsDTO} object containing details of Interest rate return by Pricing module. This rate will be used for interest computation. @param baseForInterest TODO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromFact	This method will derive Base for Interest calculation which is based on Fact.  Fact should have datatype as Money.  Value of derived fact will be returned to caller.  @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.	This method will decide which call to make : call the rules engine to execute a rule / expression or make a call to the fact derivation service with the fact id. Since both the values will not be present at the same time, it is required to make only one of the two calls at a time. The get value method will respond with the actual value of the balance base to the caller. The Algorithm is as Follows: 1.Get the instance of InterestPricingAdapterFactory 2.Using the instance call the getAdapter method to get InterestPricingAdapter 3.Check for the baseDerivationType type and call the appropriate method of InterestPricingAdapter a. if baseDerivationType=FACT call deriveBalanceBaseFromFact b. if baseDerivationType=RULE call deriveBalanceBaseFromRule c. if baseDerivationType=EXPRESSION call deriveBalanceBaseFromExpression
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromFact	This method will derive Base for Interest calculation which is based on Fact.  Fact should have datatype as Money.  Value of derived fact will be returned to caller.  @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.	This method will derive Base for Interest calculation which is based on Fact.  It will call {@link FactDerivationService} to derive value of fact which will be used as base for interest calculation.  Following is the algorithm for this method:  <ol><li>Start fact derivation session by calling startSession method {@link FactDerivationService}.</li> <li>Call deriveFactValue method in {@link FactDerivationService} to get value of fact.</li> <li>Close fact derivation session by calling closeSession method {@link FactDerivationService}.</li> <li>Return derived fact value to the caller.</li> </ol> @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromFact	This method will derive Base for Interest calculation which is based on Fact.  Fact should have datatype as Money.  Value of derived fact will be returned to caller.  @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.	This method will derive Base for Interest calculation which is based on Rule.  It will call {@link RuleExecutionApplicationService} to derive value of expression which will be used as base for interest calculation.  Following is the algorithm for this method:  <ol> <li>Retrieve {@link SessionContext} from {@link FCRTThreadAttribute}.</li> <li>Create {@link RuleContext} using rule id, {@link FactContext} and domain and domain categories present inside {@link FactContext}.</li> <li>Call executeRule method in {@link RuleExecutionApplicationService} using parameters {@link SessionContext} and {@link RuleContext} to execute rule.</li> <li>Return output of rule execution to the caller.</li> </ol> @param ruleId Identifier of rule on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Rule.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromFact	This method will derive Base for Interest calculation which is based on Fact.  Fact should have datatype as Money.  Value of derived fact will be returned to caller.  @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.	
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromFact	This method will derive Base for Interest calculation which is based on Fact.  Fact should have datatype as Money.  Value of derived fact will be returned to caller.  @param factId Identifier of fact on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Fact.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromExpression	This method will derive Base for Interest calculation which is based on Expression.  Expression should have datatype as Money.  Value of evaluated expression will be returned to caller.  @param expressionId Identifier of expression on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Expression.	This method will decide which call to make : call the rules engine to execute a rule / expression or make a call to the fact derivation service with the fact id. Since both the values will not be present at the same time, it is required to make only one of the two calls at a time. The get value method will respond with the actual value of the balance base to the caller. The Algorithm is as Follows: 1.Get the instance of InterestPricingAdapterFactory 2.Using the instance call the getAdapter method to get InterestPricingAdapter 3.Check for the baseDerivationType type and call the appropriate method of InterestPricingAdapter a. if baseDerivationType=FACT call deriveBalanceBaseFromFact b. if baseDerivationType=RULE call deriveBalanceBaseFromRule c. if baseDerivationType=EXPRESSION call deriveBalanceBaseFromExpression
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromExpression	This method will derive Base for Interest calculation which is based on Expression.  Expression should have datatype as Money.  Value of evaluated expression will be returned to caller.  @param expressionId Identifier of expression on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Expression.	This method will derive Base for Interest calculation which is based on Expression.  It will call {@link ExpressionEvaluationApplicationService} to derive value of expression which will be used as base for interest calculation.  Following is the algorithm for this method:  <ol> <li>Retrieve {@link SessionContext} from {@link FCRTThreadAttribute}.</li> <li>Call evaluateExpression method in {@link ExpressionEvaluationApplicationService} to evaluate expression.</li> <li>Return evaluated expression value to the caller.</li> </ol> @param expressionId Identifier of expression on which balance for interest is based. @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Derived Balance Base for Interest which is based on Expression.
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromExpression	This method will derive Base for Interest calculation which is based on Expression.  Expression should have datatype as Money.  Value of evaluated expression will be returned to caller.  @param expressionId Identifier of expression on which balance for interest is based.  @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Expression.	
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	deriveBalanceBaseFromRule	This method will derive Base for Interest calculation which is based on Rule.  Rule outcome should have datatype as Money.  Value of rule outcome will be returned to caller.  @param ruleId Identifier of rule on which balance for interest is based.  @param factContext {@link FactContext} is a context which stores information about interacting module with reference to Fact Vocabulary. return Derived Balance Base for Interest which is based on Rule.	This method will decide which call to make : call the rules engine to execute a rule / expression or make a call to the fact derivation service with the fact id. Since both the values will not be present at the same time, it is required to make only one of the two calls at a time. The get value method will respond with the actual value of the balance base to the caller. The Algorithm is as Follows: 1.Get the instance of InterestPricingAdapterFactory 2.Using the instance call the getAdapter method to get InterestPricingAdapter 3.Check for the baseDerivationType type and call the appropriate method of InterestPricingAdapter a. if baseDerivationType=FACT call deriveBalanceBaseFromFact b. if baseDerivationType=RULE call

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				deriveBalanceBaseFromRule c. if baseDerivationType=EXPRESSION call deriveBalanceBaseFromExpression
InterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestPricingAdapter	getFeesForArrangementNumberforPeriod	This method will fetch the total amount of fees charged for an arrangement number for the input period. @param arrangementNumber Arrangement Number for which fees have to be fetched. @param startDate Start date of the period. @param endDate End Date of the period. @return @throws FatalException @throws FatalException	This method is called by the consumers of the Interest Module to get the Interest Earned/Paid by the Customer for a particular arrangementNumber in a given period.  Along with the Interest amount this method will also return the tax amount paid by the customer for the given arrangementNumber. For the given date range if the customer has not earned/paid any interest then the response send by this method will contain zeroMoney.  For the given interest earned by the customer :-  <style type="text/css"> table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} </style> <table class="tableizer-table"> <tr class="tableizer-firstrow"> <th>ArrangementNumber</th> <th>SettlementDate</th> <th>InterestClass</th> <th>RoundedPreTaxInterest</th> <th>RoundedPostTaxInterest</th> </tr> <tr> <td>123456789</td> <td>31-Jan-10</td> <td>Credit</td> <td>100</td> <td>90</td> </tr> <tr> <td>123456789</td> <td>31-Jan-10</td> <td>Debit</td> <td>0</td> <td>0</td> </tr> <tr> <td>123456789</td> <td>28-Feb-10</td> <td>Credit</td> <td>80</td> <td>70</td> </tr> <tr> <td>123456789</td> <td>28-Feb-10</td> <td>Debit</td> <td>0</td> <td>0</td> </tr> </table> The response send by this method :-  <style type="text/css"> table.tableizer-table {border:

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;StartDate&lt;/th&gt; &lt;th&gt;EndDate&lt;/th&gt; &lt;th&gt;netCreditPreTaxAmount&lt;/th&gt; &lt;th&gt;netCreditPostTaxAmount&lt;/th&gt; &lt;th&gt;netTaxAmount&lt;/th&gt; &lt;th&gt;netDebitAmount&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1-Jan-10&lt;/td&gt; &lt;td&gt;28-Feb-10&lt;/td&gt; &lt;td&gt;180&lt;/td&gt; &lt;td&gt;160&lt;/td&gt; &lt;td&gt;20&lt;/td&gt; &lt;td&gt;0&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param sessionContext @param accountId @param depositNumber @param startDate @param endDate @return InterestDetailsResponse @throws FatalException </pre>
IIInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IIInterestPricingAdapter	getFeesForArrangementNumberforPeriod	<p>This method will fetch the total amount of fees charged for an arrangement number for the input period. @param arrangementNumber Arrangement Number for which fees have to be fetched. @param startDate Start date of the period. @param endDate End Date of the period. @return @throws FatalException @throws FatalException</p>	<p>This method will fetch the total amount of fees charged for an arrangement number for the input period. This method calls the {@link IIInterestPricingAdapter#getFeesForArrangementNumberforPeriod(String, Date, Date)} method. @param arrangementNumber @param startDate @param endDate @param currency @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	getFeesForArrangementNumberforPeriod	This method will fetch the total amount of fees charged for an arrangement number for the input period. @param arrangementNumber Arrangement Number for which fees have to be fetched. @param startDate Start date of the period. @param endDate End Date of the period. @return @throws FatalException @throws FatalException	This method will fetch the total amount of fees charged for an arrangement number for the input period. @param arrangementNumber Arrangement Number for which fees have to be fetched. @param startDate Start date of the period. @param endDate End Date of the period. @return @throws FatalException @throws FatalException
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	getFeesForArrangementNumberforPeriod	This method will fetch the total amount of fees charged for an arrangement number for the input period. @param arrangementNumber Arrangement Number for which fees have to be fetched. @param startDate Start date of the period. @param endDate End Date of the period. @return @throws FatalException @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestPricingAdapter	getFactsUsedInExpression	This method returns Set of Identifiers of Facts used in {@link IExpression} @param expressionId @return @throws FatalException	This method is called by the consumers of the Interest module to inquire where Unclear balance is used in {@link BaseForInterest}  If the BaseForInterest of the {@link InterestRule} contains Unclear balance as a Fact and this IRD linked to the product will   become Collected balance product else will become BookBalance product.  The balance base may be derived directly from a fact or it may have to process some rule or expression to get the value. Both expressionId and factId cannot be present at the same time, only one has to be present at a time. This method returns the true if the unclear balance is used to derive the balance base for interest.   Algorithm : <ol> <li>Prepare InterestRuleKey using InterestRuleKeyDTO <li>Fetch {@link InterestRule} corresponding to {@link InterestRuleKey} using {@link InterestRuleResolutionService#fetchInterestRule()} method.</li><li>Set constituentBaseForInterest = interestRule.getConstituentBaseForInterest()</li><li>Iterate through the set to get the baseForInterest for the different constituent types attached with the {@link InterestRule} <ol> <li>If the {@link BaseDerivationType}==FACT  Check the balance base id if it contains the unclear balance as a fact. If it contains unclear balance fact then return true. <li>If the {@link BaseDerivationType} == EXPRESSION   Call {@link IIInterestPricingAdapter#getFactsUsedInExpression(expressionId)} to get the list of Facts used in Expression.  If the list of Facts returned by above method contains unclear balance fact then return true. </ol> </ol> @param sessionContext @param interestRuleKeyDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	getFactsUsedInExpression	This method returns Set of Identifiers of Facts used in <a href="#">{@link IExpression}</a> @param expressionId @return @throws FatalException	This method is called by the consumers of the Interest module to inquire where offset balance is used in <a href="#">{@link BaseForInterest}</a> This will be used to determine whether Principal Offset Facility is applicable. The balance base may be derived directly from a fact or it may have to process some rule or expression to get the value. Both expressionId and factId cannot be present at the same time, only one has to be present at a time. This method returns the true if the unclear balance is used to derive the balance base for interest. Algorithm : <ol style="list-style-type: none"> <li>Prepare InterestRuleKey using InterestRuleKeyDTO</li> <li>Fetch <a href="#">{@link InterestRule}</a> corresponding to <a href="#">{@link InterestRuleKey}</a> using <a href="#">{@link InterestRuleResolutionService#fetchInterestRule()} method.</a></li> <li>Set constituentBaseForInterest = interestRule.getConstituentBaseForInterest()</li> <li>Iterate through the set to get the baseForInterest for the different constituent types attached with the <a href="#">{@link InterestRule}</a></li> <li>If the <a href="#">{@link BaseDerivationType}==FACT</a>  Check the balance base id if it contains the offset balance as a fact. If it contains unclear balance fact then return true.</li> <li>If the <a href="#">{@link BaseDerivationType} == EXPRESSION</a>  Call <a href="#">{@link IInterestPricingAdapter#getFactsUsedInExpression(expressionId)}</a> to get the list of Facts used in Expression.  If the list of Facts returned by above method contains offset balance fact then return true.</li> </ol> @param sessionContext @param interestRuleKeyDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	getFactsUsedInExpression	This method returns Set of Identifiers of Facts used in {@link IExpression} @param expressionId @return @throws FatalException	This method returns Set of Identifiers of Facts used in {@link IExpression} @param expressionId @return @throws FatalException
IInterestPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IInterestPricingAdapter	getFactsUsedInExpression	This method returns Set of Identifiers of Facts used in {@link IExpression} @param expressionId @return @throws FatalException	
IPricingContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingContentAdapter	sendNotificationToContentManagement	This method will call content management services generateDocument and deliverDocumentForParty to generate and deliver document. @param sessionContext @param arrangementPriceChangeCode @param arrangementNumber @param partyId @param priceEffectiveDate	This method will call content management services generateDocument and deliverDocumentForParty to generate and deliver document. @param sessionContext @param arrangementPriceChangeCode @param arrangementNumber @param partyId @param priceEffectiveDate
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getContactPreference	Used to fetch party contact detail from Party	This method sets the dispatch address for alert to be send. First check for contact information maintain as Preferred one. If no Preferred address is maintain then take the personal dispatch address and send. If personal is also not maintain then use work email id. @throws FatalException @see com.ofss.fc.app.party.service.contact.IContactPreferenceApplicationService#fetchAllContactPreferences(SessionContext sessionContext, String partyId, String roleId)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getContactPreference	Used to fetch party contact detail from Party	Fetches contact preference. @param partyId @param destinationType @return @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getContactPreference	Used to fetch party contact detail from Party	
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getSessionContext		(non-Javadoc) @return SessionContext @throws FatalException @see com.ofss.fc.app.party.service.contact.IContactPreferenceApplicationService#getSessionContext()
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getSessionContext		This method process the subscribes alerts. @see com.ofss.fc.domain.ep.service.action.ActionHandlerService#processAction(com.ofss.fc.entity.integration.ep.action.IAction, com.ofss.fc.framework.domain.ep.dto.ActivityData, com.ofss.fc.entity.integration.ep.action.subscriber.ISubscriber, com.ofss.fc.entity.integration.ep.action.IActionTemplate)
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getSessionContext		Processes unsubscribe action. @see com.ofss.fc.domain.ep.service.action.ActionHandlerService#processAction(com.ofss.fc.entity.integration.ep.action.IAction, com.ofss.fc.framework.domain.ep.dto.ActivityData, com.ofss.fc.entity.integration.ep.action.subscriber.ISubscriber, com.ofss.fc.entity.integration.ep.action.IActionTemplate)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getSessionContext		This method is used to register a single event for the activity represented by the activity log. @fcb.param MI,String,activityDataId, , This is the unique identifier corresponding to the of the activity instance and data logged for event generation & further processing. @fcb.param MI,Event,event, , This parameter represents the event to be generated and mapped against the activity instance. @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getCustomerName		
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getCustomerName		This represents the fetchSecuredMail method of the entity. It returns the complete detail of the mail message based on the mail reference number and sender id given in the securedMailDTO. @param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param MI,SecuredMailDTO,securedMailDTO, ,securedMailDTO securedMailDTO captures the details of the mail message like receiver id, sender id, mail reference number, send date etc. Mandatory inputs are: mail reference number and sender id. @param MI,boolean, inboxView set inboxView as true if the mail is being fetched for inbox. @return {@link com.ofss.fc.SecuredMailInquiryResponse.nab.app.desktop.mail.SecuredMailInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for Transaction Failure section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getCustomerName		<p>This represents the fetchSecuredMailWithReferenceNo method of the entity. It returns the complete detail of the mail message based on the mail reference number.</p> <p>@param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@param MI,String,mailRefNo, This parameter defines the unique mail reference number of the mail. @return {@link com.ofss.fc.SecuredMailInquiryResponse.nab.app.desktop.mail.SecuredMailInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getCustomerName		<p>This represents the fetchSecuredMailWithReferenceNo method of the entity. It returns the complete detail of the mail message based on the mail reference number.</p> <p>@param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@param MI,String,mailRefNo, This parameter defines the unique mail reference number of the mail. @return {@link com.ofss.fc.SecuredMailInquiryResponse.nab.app.desktop.mail.SecuredMailInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getCustomerName		
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getPartyDetails	This is adapter method which fetches all associated partys from given relation and source party ids. @param relation @param sourcePartyId @return	This is adapter method which fetches all associated party's from given relation and source party ids. @param relation @param sourcePartyId @return
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getPartyDetails	This is adapter method which fetches all associated partys from given relation and source party ids. @param relation @param sourcePartyId @return	This is adapter method which fetches details for party to account relations. @param context @param relation @param sourcePartyId @param accountId @return
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getPartyDetails	This is adapter method which fetches all associated partys from given relation and source party ids. @param relation @param sourcePartyId @return	Gets the getAssociatedPartyDetails. @param rmt @param sourcePartyId @param accountId @return @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	getPartyToAccountDetails	This is adapter method which fetches all associated partys from given relation and source party ids. @param relation @param sourcePartyId @return	Gets the getAssociatedPartyDetails. @param rmt @param sourcePartyId @param accountId @return @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	checkWarningIndicatorForAlertAndNotification	Checks warning indicator applied on party or not. @param partyId @return @throws FatalException	Adapter to check the warning indicator applied on party or not. @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	checkWarningIndicatorForAlertAndNotification	Checks warning indicator applied on party or not. @param partyId @return @throws FatalException	Gets the party case details for customer or guaranter. @param rmt @param sourcePartyId @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	checkWarningIndicatorForAlertAndNotification	Checks warning indicator applied on party or not. @param partyId @return @throws FatalException	Gets the getAssociatedPartyDetails. @param rmt @param sourcePartyId @param accountId @return @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	checkWarningIndicatorForAlertAndNotification	Checks warning indicator applied on party or not. @param partyId @return @throws FatalException	Gets the getAssociatedPartyDetails. @param rmt @param sourcePartyId @param accountId @return @throws FatalException
IPartyEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IPartyEventProcessingAdapter	checkWarningIndicatorForAlertAndNotification	Checks warning indicator applied on party or not. @param partyId @return @throws FatalException	
IAssetClassificationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationPartyAdapter	fetchPartyDetails	This method returns the Party details for a given Party ID	This method triggers accounting event, updates Written Off recovered amount and inserts Written Off recovery
IAssetClassificationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationPartyAdapter	fetchPartyDetails	This method returns the Party details for a given Party ID	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationPartyAdapter	fetchPartyDetails	This method returns the Party details for a given Party ID	
IAssetClassificationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationPartyAdapter	getLegalNameOfParty	This method fetches the name of the party, from a given partyId. @param context @param partyId @return String @throws FatalException	@param accountId @param partyId @param reason1 @param processDate @param totalWrittenOffAmount @return
IDeleteOrphanChildrenAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\IDeleteOrphanChildrenAdapter	deleteOrphanChildren		This represents the updateCondition method of the entity. @param collateral
IDeleteOrphanChildrenAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\IDeleteOrphanChildrenAdapter	deleteOrphanChildren		This method is used to update the collateral as a result of the party merge. A separate method is being added so that the facility related calls are not done. @param collateral @throws FatalException
IAccountWriteOffProcessCallingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IAccountWriteOffProcessCallingAdapter	callAccountWriteOffProcess	@param accountNumber @return TransactionStatus @throws FatalException	@param accountNumber @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
ISDNCheckHistoryApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISDNCheckHistoryApplicationServiceAdapter	fetchSDNCheckStatus		This method checks the SDN status of the logged in party   If the logged in user has his status set as SDN,the method  triggers a fatal exception.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISDNCheckHistoryApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISDNCheckHistoryApplicationServiceAdapter	fetchSDNCheckStatus		
ICardRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardRulesAdapter	executeRuleForBillCode	This method executes the rule and returns the credit card bill code details @param sessionContext @param ruleId @param factDTOs @return CreditCardBillCodeDTO @throws FatalException	The method fetchBillCode is used to fetch the bill code based on the derivation fact provided.  @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BillCodeDerivationRequestDTO,billCodeDerivationRequestDTO, This attribute contains account id, credit card account configuration dto and bundle id. @return {@link CreditCardBillCodeResponse} This class contains the information related to credit card account details. @throws FatalException
ICardRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardRulesAdapter	executeRuleForBillCode	This method executes the rule and returns the credit card bill code details @param sessionContext @param ruleId @param factDTOs @return CreditCardBillCodeDTO @throws FatalException	This method executes the rule and returns the credit card bill code details @param sessionContext @param ruleId @param factDTOs @return CreditCardBillCodeDTO @throws FatalException
ICardRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardRulesAdapter	executeRuleForBillCode	This method executes the rule and returns the credit card bill code details @param sessionContext @param ruleId @param factDTOs @return CreditCardBillCodeDTO @throws FatalException	
IAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	getAlertDisplayPartyDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	getAlertDisplayPartyDtls		This method will be used by the consulting team.
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	getAlertDisplayPartyDtls		This method fetches activity log for outgoing collection dishonour alerts. @param debtorBank @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode @param instructionNumber @param counterpartyName @param counterpartyBank @param txnAmount @param transactionDate @param rejectReason @param productId @return
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	fetchJointName		
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	sendDetailsChangedNotification		Service used by AIA to update OBP party data model
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	sendDetailsChangedNotification		
IAlertUtility Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\alert\IAlertUtilityAdapter	generateNotificationWidget		This method will be used by the consulting team.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	fetchNotificationDetails		This method will be used by the consulting team.
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	getAlertDisplayPartyDetails		
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	getAlertDisplayPartyDetails		This method will be used by the consulting team.
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	getAlertDisplayPartyDetails		This method fetches activity log for outgoing collection dishonour alerts. @param debtorBank @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode @param instructionNumbar @param counterpartyName @param counterpartyBank @param txnAmount @param transactionDate @param rejectReason @param productId @return
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	fetchPayeeNames		This method will be used by the consulting team.
IAAlertUtilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IAAlertUtilityAdapter	formatAmount		This method will be used by the consulting team.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchTDA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchTDAapt er	fetchTDAcc ountDetails	<p>This method returns the TermDepositNotificaitonResponse that consist of various things like maturity amount, maturity date, principal amount etc. @param accountNumber @return the <tt>TermDepositNotificaitonRespon se</tt> object. @throws FatalException	<p>This method will call the BranchTDAadapter to fetch the deposit details for a Term Deposit Account. details like maturity amount, principal amount, deposit status, maturity date. </p> @param accountNumber @return @throws FatalException
IBranchTDA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchTDAapt er	fetchTDAcc ountDetails	<p>This method returns the TermDepositNotificaitonResponse that consist of various things like maturity amount, maturity date, principal amount etc. @param accountNumber @return the <tt>TermDepositNotificaitonRespon se</tt> object. @throws FatalException	
IBranchTDA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchTDAapt er	fetchTermB asedOnAcc ountAndDe positID	<p>This method returns the DepositResponse that consist of various things like maturity amount, interst payout frequency maturity date, principal amount etc. @param tdAccount @param depositID @return the <tt>DepositResponse</tt> object. @throws FatalException	
IDDAWorkfl owAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkflowA dapter	createWork flowTracker Summarize dEntity	@param workflowTrackerSummarizedEntity DTO	This method is designed to raise a request for Account Closure. For further details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosure ApplicationService#createClosureRequest(SessionCont ext, DDAClosureRequestDTO)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	createWorkflowTrackerSummarizedEntity	@param workflowTrackerSummarizedEntity DTO	This method will create the workflow tracker summarized entity. @param sessionContext @param ddaClosurerequestDTO
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	createWorkflowTrackerSummarizedEntity	@param workflowTrackerSummarizedEntity DTO	
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the summary text and the closure status in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method is designed to update the details about the Account Closure Request, which is already being raised. This method is invoked when negotiation with the customer regarding closing account / cancellation of the same is ended. The request details are updated with the RetentionStrategy defined by Operation Executive. For further details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosureApplicationService#updateClosureRequest(SessionContext, DDAClosureRequestDTO)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the summary text and the closure status in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	<p>TODO:ATM card needs to be marked as destroyed. Amit to get back where this status is maintained. This method will be used for the final closure of the account. It will settle the interest components, fees, tax, process the closure withdrawal and update the account status to Closed. The method works as follows :</p> <ul style="list-style-type: none"> <li>Validate whether the account details are permissible for account closure or not by calling the domain service {@link DDAClosureService#validateAccountClosure}. The value of the parameter isAccountBeingClosed will be passed as True.</li> <li>Call the method {@link DDAClosureService#processFeePayment} to post the fee entries due for the account and update the account balance</li> <li>Call the method {@link DDAClosureService#processInterestAndTaxSettlement} to compute the interest and pass accrual and capitalization entries.</li> <li>Call the method {@link DDAClosureService#markUnpaidChequesAsDestroyed} to update the status of all the unused cheques, so that they cannot be issued by the account holder once the account is closed.</li> <li>If the input.closureRequestId is null, it implies that the screen is being used standalone Call the method {@link DemandDepositAccountService#modifyClosureReasons} to update the closure reasons. Call the method {@link IDDASettlementAdapter#executePreferenceOnline} with the settlementAmount returned by the service {@link DDAClosureService#validateAccountClosure} to pay the credit amount to the customer in the mode chosen by him. Else Call the service {@link DDAClosureService#fetchClosureRequest} to inquire the request details and retrieve the</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>settlementReferenceNumber.</li> <li>Call the method <a href="#">IDDASettlementAdapter#processSettlement</a> with the settlementAmount returned by the service <a href="#">DDAClosureService#validateAccountClosure</a> to pay the credit amount to the customer in the mode chosen by him.</li> <li>Delete the future dated events for the account by calling the service <a href="#">ActionLogger#removeAllIDDAActions</a></li> <li>Unregister the pricing plan linked to the account by calling the service <a href="#">IDDA PricingAdapter#closePricingPlan</a></li> <li>Unregister the interest plan linked to the account by calling the service <a href="#">IDDAInterestAdapter#unregisterPlan</a></li> <li>Finally call the method <a href="#">AccountStatusManagementService#modifyAccountStatus</a> to mark the account as closed.</li> <li>Check the overridable conditions by calling the method <a href="#">OverridableFinancialBusinessPolicy#validate</a></li> </ul> <p>@param accountId @param closureDTO @throws FatalException</p>
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the summary text and the closure status in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the workflow tracker summarized entity. @param ddaClosurerequestDTO
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the summary text and the closure status in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the workflow tracker summarized entity. @param ddaClosureRequestDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkflowAdapter	updateWorkflowTrackerSummarizedEntity	This method will update the summary text and the closure status in the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	
ITDEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method logs the activity for Financial events. These will then be forwarded to the party as notification through SMS/Email for events @param activityId @param eventId @param accountId @param depositId @param transactionDetail
ITDEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method logs the activity for Non-Financial events. These will then be forwarded to the party as notification through SMS/Email for events @param activityId @param eventId @param accountId @param depositId @throws FatalException
ITDEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method persists activity and event log for given activity and event id @param activityId @param eventId @param activityLog @param eventDate @throws FatalException
ITDEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method logs the activity for Redemption Financial events. These will then be forwarded to the party as notification through SMS/Email for events @param activityId @param eventId @param accountId @param depositId @param redemptionDetail
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	doTDRedemption	The following method is used to perform td redemption @param sessionContext @param redeemTDRequest @return @throws FatalException	The following method is used to execute cash withdrawal for TD account. @param contextResponse @param businessContext @param taskWithdraw @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	doTDRedemption	The following method is used to perform td redemption @param sessionContext @param redeemTDRequest @return @throws FatalException	The following method is used to perform TD transaction when the account in context is a TD account. Fund new TD and Redeem New Td actions are performed based on the Account in context and the direction supplied. @param businessContext @param dashboardService @param fromAccount @param toAccount @param amount @param depositNo @param isInflowTransaction @param settlementModeType @throws FatalException
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	doTDRedemption	The following method is used to perform td redemption @param sessionContext @param redeemTDRequest @return @throws FatalException	The following method is used to perform TD transaction when caller account is a TD account. The method is called when the account in context in a TD account. @param businessContext - will have account type as Term Deposit @param dashboardService @param fromAccount @param toAccount @param amount @param depositNo @param isInflowTransaction @param settlementModeType @throws FatalException
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	doTDRedemption	The following method is used to perform td redemption @param sessionContext @param redeemTDRequest @return @throws FatalException	The following method is used to do td redemption.
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	createNewTermDeposit	The following method is used to create new term deposit @param sessionContext @param tdPayinRequestDTO @return @throws FatalException	The following method is used to perform TD transaction when the account in context is a TD account. Fund new TD and Redeem New Td actions are performed based on the Account in context and the direction supplied. @param businessContext @param dashboardService @param fromAccount @param toAccount @param amount @param depositNo @param isInflowTransaction @param settlementModeType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	createNewTermDeposit	The following method is used to create new term deposit @param sessionContext @param tdPayinRequestDTO @return @throws FatalException	The following method is used to perform TD transaction when caller account is a TD account. The method is called when the account in context in a TD account. @param businessContext - will have account type as Term Deposit @param dashboardService @param fromAccount @param toAccount @param amount @param depositNo @param isInflowTransaction @param settlementModeType @throws FatalException
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	createNewTermDeposit	The following method is used to create new term deposit @param sessionContext @param tdPayinRequestDTO @return @throws FatalException	The following method is used to create new td
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	createNewTermDeposit	The following method is used to create new term deposit @param sessionContext @param tdPayinRequestDTO @return @throws FatalException	The following method is used to execute deposit cash command. Denomination is required only when denomination tracking is enabled.
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	fetchAllDeposits	The following method is used to fetch all deposits @param sessionContext @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	<p>This method is used to inquire details required to perform redemption operation&lt;br&gt;This method will be directly called by UI, to inquire redemption details.</p> <p>@param sessionContext This parameter captures the basic information about the user of the system. @param redemptionDetailsInquiryRequestDTO {@link RedemptionRequestDTO} Object containing details required to get redemption details @return {@link RedemptionDetailsDTO} Object containing details of redemption</p> <p>@see com.ofss.fc.app.td.service.transaction.redemption.ITermDepositRedemptionApplicationService#inquireRedemptionDetails(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.redemption.RedemptionRequestDTO)</p>	The following method is used to view the deposit details after full redemption.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	<p>This method is used to inquire details required to perform redemption operation&lt;br&gt;This method will be directly called by UI, to inquire redemption details.</p> <p>@param sessionContext This parameter captures the basic information about the user of the system. @param redemptionDetailsInquiryRequestDTO {@link RedemptionRequestDTO} Object containing details required to get redemption details @return {@link RedemptionDetailsDTO} Object containing details of redemption</p> <p>@see com.ofss.fc.app.td.service.transaction.redemption.ITermDepositRedemptionApplicationService#inquireRedemptionDetails(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.redemption.RedemptionRequestDTO)</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	<p>This method is used to inquire details required to perform redemption operation&lt;br&gt;This method will be directly called by UI, to inquire redemption details.</p> <p>@param sessionContext This parameter captures the basic information about the user of the system. @param redemptionDetailsInquiryRequestDTO {@link RedemptionRequestDTO} Object containing details required to get redemption details @return {@link RedemptionDetailsDTO} Object containing details of redemption</p> <p>@see com.ofss.fc.app.td.service.transaction.redemption.ITermDepositRedemptionApplicationService#inquireRedemptionDetails(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.redemption.RedemptionRequestDTO)</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	<p>This is the light API which is used to inquire redemption details.&lt;br&gt;This API provides details like remaining principal balance after redemption, net proceed, net interest payable, tax deducted and penalty applicable in case of premature redemption. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Account id of term deposit @param depositId Term deposit number @param redemptionAmount Money object containing redemption amount @param redemptionAmountType Type of redemption whether full or part @return {@link RedemptionInquiryDetailsResponse} Object containing redemption inquiry details @throws FatalException @see {@link ITermDepositBasicApplicationService#inquireRedemptionDetails(SessionContext, String, Integer, MoneyDTO)}</p>	The following method is used to view the deposit details after full redemption.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	<p>This is the light API which is used to inquire redemption details.&lt;br&gt;This API provides details like remaining principal balance after redemption, net proceed, net interest payable, tax deducted and penalty applicable in case of premature redemption. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Account id of term deposit @param depositId Term deposit number @param redemptionAmount Money object containing redemption amount @param redemptionAmountType Type of redemption whether full or part @return {@link RedemptionInquiryDetailsResponse} Object containing redemption inquiry details @throws FatalException @see {@link ITermDepositBasicApplicationService#inquireRedemptionDetails(SessionContext, String, Integer, MoneyDTO)}</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	inquireRedemptionDetails	This is the light API which is used to inquire redemption details. This API provides details like remaining principal balance after redemption, net proceed, net interest payable, tax deducted and penalty applicable in case of premature redemption. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Account id of term deposit @param depositId Term deposit number @param redemptionAmount Money object containing redemption amount @param redemptionAmountType Type of redemption whether full or part @return {@link RedemptionInquiryDetailsResponse} Object containing redemption inquiry details @throws FatalException @see {@link ITermDepositBasicApplicationService#inquireRedemptionDetails(SessionContext, String, Integer, MoneyDTO)}	
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	fetchDepositAmount	This method is used to fetch the deposit details. @param accountId Account id of term deposit @param depositId Term deposit id @return {@link Deposit} Fetched deposit object	The following method is used to fetch the deposit amount. @param accountId @param depositId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLTermDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLTermDepositAdapter	fetchDepositAmount	This method is used to fetch the deposit details. @param accountId Account id of term deposit @param depositId Term deposit id @return {@link Deposit} Fetched deposit object	The following method is used to fetch the RedemptionAmountType using amount. @param accountId @param depositId @param redemptionAmount @return
IWorkflowManagerCallingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IWorkflowManagerCallingAdapter	abortCompositeInstancesByTitle	@param WorkflowManagerWrapperDTO @return boolean @throws FatalException	@param WorkflowManagerWrapperDTO @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
IPricingSMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingSMSAdapter	addAuthorizationConfigurationsForFee		
IPricingSMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingSMSAdapter	addAuthorizationConfigurationsForFee		This method will publish PPC configuration for fees. @param sessionContext @param pricePolicyChart @throws FatalException
IFraudAssertionSessionIDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionSessionIDAdapter	getSessionId	This method will generate OAAM session Id, which is required for OAAM call execution for fraud check. @return String - OAAM session ID required for fraud check @throws FatalException	This method will generate OAAM session Id, which is required for OAAM call execution for fraud check. @return String - OAAM session ID required for fraud check @throws FatalException
IMDMInquireAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\IMDMInquireAdapter	populateData		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ChannelAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\ChannelAccountAdapter	fetchProductType	This method fetches the product code, which is then used for performing limit validation. @param sessionContext @param accountId @return productType @throws FatalException	This method returns the instance of channel adapter. @return ChannelAccountAdapter class instance
ChannelAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\ChannelAccountAdapter	fetchProductType	This method fetches the product code, which is then used for performing limit validation. @param sessionContext @param accountId @return productType @throws FatalException	This method returns product code. @param sessionContext @param accountId @return String @throws FatalException
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	modifyUsingCounterpartyDetails	The following method is used to modify the counterparty details. @param objectValue @param partyId @throws FatalException	The following method is used to modify the component Preference based on the counter party that is being supplied. @param componentPreferenceDTO @throws FatalException
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	modifyUsingCounterpartyDetails	The following method is used to modify the counterparty details. @param objectValue @param partyId @throws FatalException	
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	modifyUsingCounterpartyDetails	The following method is used to modify the counterparty details. @param objectValue @param partyId @throws FatalException	
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	modifyUsingCounterpartyDetails	The following method is used to modify the counterparty details. @param objectValue @param partyId @throws FatalException	This method is used to fill the ElectronicTransferCaptureDTO. @param SettlementModeDTO details @param Money amount @param String arrangementId, @param TaskModuleType moduleType @param String narration @return ElectronicTransferCaptureDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	counterPartyModifiedResponse	The following method is used to fetch the counterparty response. @param counterPartyId @return @throws FatalException	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT. It builds the component preference item by resolving counter party Id and party Id. @param counterPartyId @param partyId @param priority @param splitValue @param currency
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	counterPartyModifiedResponse	The following method is used to fetch the counterparty response. @param counterPartyId @return @throws FatalException	The following method is used to modify details in settlement mode details DTO @param settlementModeDTO @throws FatalException
ISettlementPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPaymentAdapter	counterPartyModifiedResponse	The following method is used to fetch the counterparty response. @param counterPartyId @return @throws FatalException	The following method is used to modify details in the settlement mode entity @param settlementMode @throws FatalException
ITaxationFactsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationFactsAdapter	getFactValues		This will return the {@link PricingEventContext} which is used by the pricing module. @param taxAmount @return
ITaxationFactsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationFactsAdapter	getFactValues		
ITaxationFactsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationFactsAdapter	getFactValues		This method is used to execute the rule which is created at time of rule maintenance. This method will internally call {@link ITaxationRuleApplicationAdapter} adapter to execute the rule by passing rule id, facts and process date as inputs. @param ruleID @param partyInformation



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param processDate @return ArrayList<OutcomeDetails>
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	publishRule	This method will publish the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will maintain a covenant policy record. A covenant policy has 2 elements as   part of the key elements i.e the policy identifier and the effective date combination.   One covenant policy can have 1 or more effective date records maintained and active.   If a record for the combination of covenant policy identifier and effective date exists,  then the system will throw an exception.   For a particular effective date, one cannot define different policies with same Entity Linkage Level.  The rules service is called via the lcmRulesAdapter to persist the rule definition.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.   @fcb.param MI,CovenantPolicyDTO,covenantPolicyDTO, ,Data associated with the covenant policy   using which the system will create a record for the concerned covenant policy. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the   transaction status which contains the required information about the status of execution   indicating if the invocation succeeded or failed. @fcb.task LCMTaskConstants.MNT_LCM_COVENANT_POLICY @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	publishRule	This method will publish the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will maintain a conditions policy record. A conditions policy has 2 elements as   part of the key elements i.e the policy identifier and the effective date combination.   One conditions policy can have 1 or more effective date records maintained and active.   For a particular effective date, one cannot define different policies with same Entity Linkage Level.  The rules service is called using adapter to persist the rule definition.  During credit assessment, the system will pick the effective record of the conditions policy   for the combination of entities based on the rule defined. If a record for the combination of conditions policy   identifier and effective date exists, then the system will throw an exception.   @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Session Context which must contain the   session context information such as user id, branch, branch code, channel etc. The information is used by the system for authentication purposes. @fcb.param MI,ConditionsPolicyDTO,conditionsPolicyDTO, ,conditionsPolicyDTO, ,Data associated with the conditions policy   using which the system will create a record for the concerned conditions policy. @return {@link com.ofss.fc.service.response.TransactionStatus}An instance of the   transaction status which contains the required information about the status of execution   indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNT_LCM_CONDITIONS_POLICY @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	publishRule	This method will publish the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This service will publish the rule for given rule definition. @param sessionContext,ruleDefinition @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	publishRule	This method will publish the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will return the Rules Adapter based on "IsMockEnabled" flag. @param adapterClass
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	modifyRule	This method will modify the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will update an existing covenant policy record. This method will expect that the record for the covenant policy identifier and effective date is already created in the system. The attributes of the record like the policy name, rule definition, entity linkage level, domain id, domain category id can be modified using this method. The rules service is called via the OriginationRulesAdapter to update the rule definition. Modification of entity linkage level will be allowed only if there is no policy defined having the same entity linkage level for the same effective date. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,CovenantPolicyDTO,covenantPolicyDTO, , Data associated with the covenant policy using which the system will update the record for the concerned covenant policy. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				succeeded or failed. @fcb.task OriginationTaskConstants.MNT_LCM_COVENANT_POLICY @throws com.ofss.fc.infra.exception.FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	modifyRule	This method will modify the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will update an existing conditions policy record. This method will expect that the record for the conditions policy identifier and effective date is already created in the system. The attributes of the record like the rule definition, entity linkage level, authoring mode, domain id, domain category id can be modified using this method. The rules service is called using adapter to update the rule definition. Modification of entity linkage level will only be allowed if there is no policy defined having the same entity linkage level for the same effective date. @fcb.param MI, com.ofss.fc.app.context.SessionContext, sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. The information is used by the system for authentication purposes. @fcb.param MI, ConditionsPolicyDTO, conditionsPolicyDTO, conditionsPolicyDTO, Data associated with the conditions policy using which the system will create a record for the concerned conditions policy. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNT_LCM_CONDITIONS_POLICY @throws com.ofss.fc.infra.exception.FatalException

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	modifyRule	This method will modify the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This service will modify the existing rule for given rule id. @param sessionContext, ruleDefinition @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	modifyRule	This method will modify the rule by interacting with rules module. @param sessionContext @param ruleDefinition @throws FatalException	This method will return the Rules Adapter based on "IsMockEnabled" flag. @param adapterClass
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewRule	This method will fetch the rule by interacting with rules module. @param sessionContext @param ruleId @param effectiveDate @return @throws FatalException	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId, effectiveDate @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewRule	This method will fetch the rule by interacting with rules module. @param sessionContext @param ruleId @param effectiveDate @return @throws FatalException	This method will return the Rules Adapter based on "IsMockEnabled" flag. @param adapterClass

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewLatestRuleForProcessDate	This method will fetch the rule for process date by interacting with rules module. @param sessionContext @param ruleId @return @throws FatalException	This method will inquire a covenant policy record. A covenant policy has 2 elements as   part of the key elements i.e the policy identifier and the effective date combination.   One covenant policy can have 1 or more effective date records maintained and active.   The rules service is called via the OriginationRulesAdapter to fetch the rule definition.  Based on the policy Id and effective date inputs from the user the corresponding covenant policy   will be fetched.If a record for the combination of covenant policy   identifier and effective date does not exist, then the system will throw an exception.   @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CovenantPolicyDTO,covenantPolicyDTO, ,Data associated with the covenant policy   using which the system will inquire for a record for the concerned covenant policy. @return {@link com.ofss.fc.app.lcm.dto.core.covenant.CovenantPolicyInquiryResponse}   An instance of covenant policy which contains the attributes like rule definition,   entity linkage level, policy name, domain id,domain category id.  @fcb.task OriginationTaskConstants.MNT_LCM_COVENANT_POLICY @throws com.ofss.fc.infra.exception.FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewLatestRuleForProcessDate	This method will fetch the rule for process date by interacting with rules module. @param sessionContext @param ruleId @return @throws FatalException	This method is used by {@link #isInputCovenantCategoryLinkedToActiveCovenantPolicy(SessionContext, String)}  It fetches all the condition categories linked to all the covenant policies present in the system.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewLatestRuleForProcessDate	This method will fetch the rule for process date by interacting with rules module. @param sessionContext @param ruleId @return @throws FatalException	This method will inquire a conditions policy record. A conditions policy has 2 elements as   part of the key elements i.e the policy identifier and the effective date combination.   One conditions policy can have 1 or more effective date records maintained and active.  The rules service is called using adapter to fetch the rule definition.  Based on the policy Id and effective date inputs from the user the corresponding Conditions policy will be fetched.If a record for the combination of conditions policy   identifier and effective date does not exist, then the system will throw an exception.   @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Session Context which must contain the   session context information such as user id, branch, branch code, channel etc. The information is used by the system for authentication purposes. @fcb.param MI,ConditionsPolicyDTO,conditionsPolicyDTO, ,conditionsPolicyDTO, ,Data associated with the conditions policy using which the system will create a record for the concerned conditions policy.  @return {@link com.ofss.fc.app.lcm.dto.core.condition.ConditionsPolicyInquiryResponse}  An instance of the conditions policy which contains the attributes like rule definition,entity linkage level, authoring mode,domain id,domain category id, rule definition.  @fcb.task OriginationTaskConstants.MNT_LCM_CONDITIONS_POLICY @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewLatestRuleForProcessDate	This method will fetch the rule for process date by interacting with rules module. @param sessionContext @param ruleId @return @throws FatalException	This method is used by {@link #isInputConditionCategoryLinkedToActiveConditionPolicy(SessionContext, String)}  It fetches all the condition categories linked to all the condition policies present in the system.
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	viewLatestRuleForProcessDate	This method will fetch the rule for process date by interacting with rules module. @param sessionContext @param ruleId @return @throws FatalException	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This method accepts a ruleId, retrievalKey, retrievalKeyValue and executes the rule and returns the OutComeDetails @param ruleId @param retrievalKey @param retrievalKeyValue @return OutcomeDetails @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This service will return the Covenant Category and covenants which are linked with the entity and facts of that entity. @param SessionContext @param EntityLinkageLevelType @param String [] facts @return com.ofss.fc.app.lcm.dto.core.covenant.CovenantFactEntityInquiryResponse @throws com.ofss.fc.infra.exception.FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This service will return the Covenant Category and covenants which are linked with the entities and facts of that entity. @param SessionContext @param EntityLinkageLevelType [] @param String [] facts @return com.ofss.fc.app.lcm.dto.core.covenant.CovenantFactEntityInquiryResponse @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This method will accept session context and CollateralValuationDTO it fetches the associated ruleid from CollateralCategoryPreference and execute that rule and returns no. of Valuation required for given Collateral. @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationDTO collateralValuationDTO : Collateral valuation request details DTO. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationInquiryResponse} This class holds the values of all the DTOs defined in it.  @throws FatalException @throws RuntimeException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This represents the fetchConditionsForFactsAndEntity method. And it returns Conditions and Condition categories for the given EntityLinkageType(Party/Collateral/Product). @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,EntityLinkageLevelType,linkageType, ,EntityLinkageLevelType TODO: Explain the linkageType @fcb.param MI,String,facts, ,String TODO: Explain the facts above @return {@link com.ofss.fc.app.lcm.dto.core.condition.ConditionsForFactsAndEntityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Failure</i> section
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This represents the fetchConditionsForArrayOfFactsAndEntity method. And it returns Conditions and Condition categories for the given EntityLinkageType(Party/Collateral/Product). @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FactsAndEntityLinkageDTO,factsAndEntityLinkageDTO, ,FactsAndEntityLinkageDTO TODO: Explain the factsAndEntityLinkageDTO above @return {@link com.ofss.fc.app.lcm.dto.core.condition.ConditionsForFactsAndEntityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This service will execute the rule for given rule id and facts. @param sessionContext, ruleId, factDTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	This method will execute the rule by interacting with rules module. @param sessionContext @param ruleId @param factDTOs @return @throws FatalException	This method will return the Rules Adapter based on "IsMockEnabled" flag. @param adapterClass
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchFactsForTheRule	This method will return all the factsCodes linked to a rule @param ruleId @param effectiveDate @return facts[] @throws FatalException	This method accepts a ruleId, retrievalKey, retrievalKeyValue and executes the rule and returns the OutComeDetails @param ruleId @param retrievalKey @param retrievalKeyValue @return OutcomeDetails @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchFactsForTheRule	This method will return all the factsCodes linked to a rule @param ruleId @param effectiveDate @return facts[] @throws FatalException	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchFactsForTheRule	This method will return all the factsCodes linked to a rule @param ruleId @param effectiveDate @return facts[] @throws FatalException	This method will accept session context and CollateralValuationDTO it fetches the associated ruleid from CollateralCategoryPreference and execute that rule and returns no. of Valuation required for given Collateral. @param sessionContext @param valuationDTO @return noOfValuationRequired @throws FatalException @throws RuntimeException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchFactsForTheRule	This method will return all the factsCodes linked to a rule @param ruleId @param effectiveDate @return facts[] @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchFactsForTheRule	This method will return all the factsCodes linked to a rule @param ruleId @param effectiveDate @return facts[] @throws FatalException	This method will accept session context and CollateralValuationDTO it fetches the associated ruleid from CollateralCategoryPreference and execute that rule and returns no. of Valuation required for given Collateral. @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationDTO collateralValuationDTO : Collateral valuation request details DTO. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationInquiryResponse} This class holds the values of all the DTOs defined in it.  @throws FatalException @throws RuntimeException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This method accepts a ruleId, retrievalKey, retrievalKeyValue and executes the rule and returns the OutComeDetails @param ruleId @param retrievalKey @param retrievalKeyValue @return OutcomeDetails @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This service will return the Covenant Category and covenants which are linked with the entity and facts of that entity. @param SessionContext @param EntityLinkageLevelType @param String [] facts @return com.ofss.fc.app.lcm.dto.core.covenant.CovenantFactEntityInquiryResponse @throws com.ofss.fc.infra.exception.FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This service will return the Covenant Category and covenants which are linked with the entities and facts of that entity. @param SessionContext @param EntityLinkageLevelType [] @param String [] facts @return com.ofss.fc.app.lcm.dto.core.covenant.CovenantFactEntityInquiryResponse @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This method will accept session context and CollateralValuationDTO it fetches the associated ruleid from CollateralCategoryPreference and execute that rule and returns no. of Valuation required for given Collateral. @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationDTO collateralValuationDTO : Collateral valuation request details DTO. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationInquiryResponse} This class holds the values of all the DTOs defined in it.  @throws FatalException @throws RuntimeException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This represents the fetchConditionsForFactsAndEntity method.And it returns Conditions and Condition categories for the given EntityLinkageType(Party/Collateral/Product). @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,EntityLinkageLevelType,linkageType, ,EntityLinkageLevelType TODO: Explain the linkageType @fcb.param MI,String,facts, ,String TODO: Explain the facts above @return {@link com.ofss.fc.app.lcm.dto.core.condition.ConditionsForFactsAndEntityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Failure</i> section
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This represents the fetchConditionsForArrayOfFactsAndEntity method.And it returns Conditions and Condition categories for the given EntityLinkageType(Party/Collateral/Product). @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FactsAndEntityLinkageDTO,factsAndEntityLinkageDTO, ,FactsAndEntityLinkageDTO TODO: Explain the factsAndEntityLinkageDTO above @return {@link com.ofss.fc.app.lcm.dto.core.condition.ConditionsForFactsAndEntityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This service will execute the rule for given rule id and facts. @param sessionContext, ruleId, factDTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	executeRule	@param ruleId @param factInfoDTOArray @return OutcomeDetails	This method will return the Rules Adapter based on "IsMockEnabled" flag. @param adapterClass
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchSessionContext	@param applicationContext @return	This method accepts a ruleId, retrievalKey, retrievalKeyValue and executes the rule and returns the OutComeDetails @param ruleId @param retrievalKey @param retrievalKeyValue @return OutcomeDetails @throws FatalException
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchSessionContext	@param applicationContext @return	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchSessionContext	@param applicationContext @return	
ILCMRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\core\ILCMRulesAdapter	fetchSessionContext	@param applicationContext @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	initialize	This method will initialize the ComponentPreferenceDTO with all the basic details as well as create the settlement instructions under on complete componentPreferenceDTO @param splitType @param moduleType @param directionType @param commonSettlementDTO @param arrangementId @param partyId @param offerId	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	initialize	This method will initialize the ComponentPreferenceDTO with all the basic details as well as create the settlement instructions under on complete componentPreferenceDTO @param splitType @param moduleType @param directionType @param commonSettlementDTO @param arrangementId @param partyId @param offerId	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForLoan	This is a internal method will be called after initialize method. Basically It is being used to fill loan settlement details in case the chose settlement type is LOAN . @param priority @param splitValue @param loanAccount @param offerId @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForPaymentToExternalAccount	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT . @param priority @param splitValue @param counterPartyAccountNumber @param creditorName	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForPaymentToExternalAccount	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT . @param priority @param splitValue @param counterPartyAccountNumber @param creditorName	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForPaymentToExternalAccount	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT. It builds the component preference item by resolving counter party Id and party Id. @param counterPartyId @param partyId @param priority @param splitValue @param currency	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForPaymentToExternalAccount	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT. It builds the component preference item by resolving counter party Id and party Id. @param counterPartyId @param partyId @param priority @param splitValue @param currency	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForCollections	This is a internal method will be called after initialize method. Basically It is being used to fill collection settlement details in case the chose settlement type is COLL . @param priority @param splitValue @param payeeAccount @param creditorName	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForClearing	This is a internal method will be called after initialize method. Basically It is being used to fill collection settlement details in case the chose settlement type is CLRG . @param priority @param splitValue @param payeeAccount @param creditorName	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForCash	This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT . @param priority @param splitValue @param loanAccount @param offerId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLSettlementAd apter	addModeFo rLedger	This is a internal method will be called after initialize method. Basically It is being used to fill ledger settlement details in case the chose settlement type is GL. @param priority @param splitValue @param ledgerID	
IDSLSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLSettlementAd apter	addModeFo rExistingTe rmDeposit	This is a internal method will be called after initialize method. Basically It is being used to fill TD settlement details in case the chose settlement type is TDA. @param priority @param splitValue @param arrangementId @param depositID	
IDSLSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLSettlementAd apter	addModeFo rNewTerm Deposit	This is a internal method will be called after initialize method. Basically It is being used to fill New TD settlement details in case the chose settlement type is NTDA. @param priority @param splitValue @param depositID	
IDSLSettleme ntAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLSettlementAd apter	addModeFo rOwnSavin gsAccount	This is a internal method will be called after initialize method. Basically It is being used to fill own savings account settlement details in case the chose settlement type is DDAO. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForSavingsAccount	This is a internal method will be called after initialize method. Basically It is being used to fill savings account settlement details in case the chose settlement type is DDA. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addModeForOtherSavingsAccount	This is a internal method will be called after initialize method. Basically It is being used to fill Other savings account settlement details in case the chose settlement type is DDAT. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addMCIMode	This method is used to fill BankDraftDTO. @param context @param detailsDTO @param amount @param arrangementId @return @throws FatalException	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	fetchComponentPreference	This is a final method that will be called after setting up all the settlement instructions. this method will return the complete ComponentPreferenceDTO	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	fetchComponentPreference	This is a final method that will be called after setting up all the settlement instructions. this method will return the complete ComponentPreferenceDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	fetchComponentPreference	This is a final method that will be called after setting up all the settlement instructions. this method will return the complete ComponentPreferenceDTO	The following method is used to request for a MC.. @param contextResponse @param businessContextDTO @param taskRequestMC @param accountType @param benefName @param accountNo @param instrumentAmount @throws FatalException
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addDDAMode	This is a internal method will be called after initialize method. Basically It is being used to fill own savings account settlement details in case the chose settlement type is DDAO. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	addDDAMode	This is a internal method will be called after initialize method. Basically It is being used to fill own savings account settlement details in case the chose settlement type is DDAO. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException	
IDSLSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLSettlementAdapter	buildComponentPreferenceDTOForMCIssue		This method is used to build ComponentPreferenceDTO which is used for MC issue. @param context @param detailsDTO @param amount @param arrangementId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	Verify the account limits  1. Each account limit should be less than parent limit.  2. Each account term should be less than parent term.  3. Sum of all accounts under a facility should be less than or equal to parent limit.   4. Sum of all accounts configured under a facility of type non revolving should be less than or   equal to main facility limit and also conform to its own parent limit (same as 3 above).   5. OD Fixed Term should be less than or equal to its parent term.     Also, values for DPA (UDM, fees negotiations are read from each account configuration and set with the highest values. @param sessionContext @param creditAccountDTOs @param facilityId @param parentLimit @param parentTerm @param solutionDTO @return @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This method populates the approved amount, available amount and account title for each loan offer inside each PG in the application. @param sessionContext @param productGroupLinkage @param settlementRef @param offerReferences @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This method is used to build FacilityApprovalDataDTO
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This method performs the following validations: 1. There should not be any warning indicators for the parties applying for the PG 2. The KYC check should be completed for all the parties 3. All the preliminary conditions should have been complied with 4. Productgroup level Origination fees should have been collected 5. Application should be verified successfully @param sessionContext @param application @param prodGrpLinkage @return @throws Exception
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This will check if the commission in case of broker sourced application is overridden or not if overridden it will create approval on basis of approval configuration
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This method will fetch the the fees that are negotiated or waived during account configuration from pricing and raise overridable exception to send for approvals. @param sessionContext @param lendingProductGroupLinkageDTO @param creditFacilities @param submissionId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	This is alert send for Lending Account Creation. For account, it will take corresponding simulationID and fetch the LoanAccountConfigurationDTO. <ol><li>If PartyAccountRelationshipType is not PartyAccountRelationshipType.OTHERS send alert will be send to that party.</li> <li>If PartyAccountRelationshipType is PartyAccountRelationshipType.OTHERS then alert will be send only if for given affiliate alert to be send is allowed or not.</li> </ol> @param sessionContext @param accountOpeningDetailsDTO @param prodGroupSerialNumber @param applicationId
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendConfiguration		Verify the account limits  1. Each account limit should be less than parent limit.  2. Each account term should be less than parent term.  3. Sum of all accounts under a facility should be less than or equal to parent limit.   4. Sum of all accounts configured under a facility of type non revolving should be less than or   equal to main facility limit and also conform to its own parent limit (same as 3 above).   5. OD Fixed Term should be less than or equal to its parent term.     Also, values for DPA (UDM, fees negotiations are read from each account configuration and set with the highest values. @param sessionContext @param creditAccountDTOs @param facilityId @param parentLimit @param parentTerm @param solutionDTO @return @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendConfiguration		This method will inquire the loan Account configuration amend for the account id passed @param sessionContext @param arrangementId @return LoanAccountConfigurationDTO @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendConfiguration		This method is used to build FacilityApprovalDataDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. This method internally retrieves the following configuration saved in Loans 1)Account Preference 2)Offset Account 3)Account Payment Instruction 4)Repayment Schedule Configuration 5)Accelerated Payment Instruction 6)Rate Lock.
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		This method performs the following validations: 1. There should not be any warning indicators for the parties applying for the PG 2. The KYC check should be completed for all the parties 3. All the preliminary conditions should have been complied with 4. Productgroup level Origination fees should have been collected 5. Application should be verified successfully @param sessionContext @param application @param prodGrpLinkage @return @throws Exception
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		This will check if the commission in case of broker sourced application is overridden or not if overridden it will create approval on basis of approval configuration

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		This method will fetch the the fees that are negotiated or waived during account configuration from pricing and raise overridable exception to send for approvals. @param sessionContext @param lendingProductGroupLinkageDTO @param creditFacilities @param submissionId @return @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountAmendmentConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	updateArrangementDetails		update loan simulation with our details
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	updateArrangementDetails		This method will update loan account configuration/amend loan account configuration and update party serial number with party id
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	fetchLoanAccountInformation		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	addOutlayFunds		This method will be called for new account @param sessionContext @param prodGrpLinkage @param accountId @param LMI @param legal @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	levyFeesAndPremiumOnAccount		@param prodGrpLinkage @param nominatedAccountId @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	levyFeesAndPremiumOnAccount		This method will be called for amend account @param sessionContext @param prodGrpLinkage @param nominatedAccountId @param LMI @param legal @param origFee @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	populateDefaultLoanAccountPreferences		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	createPricingRecommendation		This API nominates an account for the facility to be nominated for fees. It performs the following on the submission <ol> <li>fetch LendingProductGroupLinkageDTO</li> <li>fetch OfferReferenceDTO</li> <li>update the isNominatedForFacilityFees flag to true</li> <li>regenerate pricingTicketNumber</li> <li>fetch CreditFacilityDTO</li> <li>create and set CreditFacilityFeeDTO into CreditFacilityDTO with updated flag for nominatedAccount which is simulationId if it is a new account else it is CreditFacilityFeeDTO.getNominatedAccount()</li></ol> @param sessionContext @param submissionId @param creditFacilityFeeDTO @param simulationId @return AccountResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	createPricingRecommendation		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	saveLoanAccountPreferenceConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	saveLoanAccountConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	inquireScheduleOnAccountConfiguration		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	populateLoanScheduleConfigurationDetails		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	openLoanAccount		This method is used to open a loan Account. @param sessionContext @param accountInputDTO @return AccountOpeningDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	openLoanAccount		
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	amendLoanAccount		This method is used to open a loan Account. @param sessionContext @param accountInputDTO @return AccountOpeningDetailsDTO @throws FatalException
IOrigination LoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLoanAdapter	saveLoanScheduleConfigurationData		
IInterestRestrictionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestRestrictionAdapter	isServiceRestricted	Responsible for checking which all services are restricted for the particular account status	This method validates whether the interest adjustment is restricted or not based on the account status. <ol> <li>If Product Class is {@link ProductClass#SAVINGS} or {@link ProductClass#CHECKING} <ol> <li>Derive the adjustment {@link Event} using {@link EventResolutionService#deriveEvent(ProductClass, InterestClass, InterestDivision, InterestConstituentType, String, AccountingAmountType, boolean)} </li><li>Prepare the {@link InterestRestrictionRequest} using the event derived in 1.1 and {@link InterestPlanItem#getInterestAgeingStatus()}</li><li>Set {@link InterestRestrictionRequest#getTransactionType()} as {@link TransactionType#Credit} if {@link InterestAdjustmentRequestDTO#getEntryType()} is equal to {@link AdjustmentAmountType#INCREASE},

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>else set it to {@link TransactionType#Debit}. &lt;li&gt;Call {@link InterestRestrictionAdapter#isServiceRestricted(InterestRestrictionRequest)} and receive the response isServiceRestricted&lt;/li&gt; &lt;li&gt;If isServiceRestricted == true , throw business exception with error code : {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED}&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;Else if Product Class is {@link ProductClass#TERM_DEPOSITS} &lt;ol&gt; &lt;li&gt;Call {@link InterestRestrictionAdapter#isAdjustmentAllowedForDeposit(String, int)} and receive the response isAllowed.&lt;/li&gt; &lt;li&gt;If isAllowed == false, throw business exception with error code : {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED_FOR_DEPOSIT}&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;/ol&gt; @param interestAdjustmentRequestDTO @param interestPlanItem @param accountId @param depositId @param interestRule</p>
InterestRestrictionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\InterestRestrictionAdapter	isServiceRestricted	Responsible for checking which all services are restricted for the particular account status	This service will check whether the interest adjustment is allowed for the Casa account.  The Casa module stores the list of restricted events for the particular account status.  If the account status is for which the interest adjustment is being performed is there in the restricted list, exception would be thrown.
InterestRestrictionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\InterestRestrictionAdapter	isAdjustmentAllowedForDeposit		This method validates whether the interest adjustment is restricted or not based on the account status.  <ol> <li>If Product Class is {@link ProductClass#SAVINGS} or {@link ProductClass#CHECKING} <ol> <li>Derive the adjustment {@link Event} using {@link EventResolutionService#deriveEvent(ProductClass, InterestClass, InterestDivision,



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>InterestConstituentType, String, AccountingAmountType, boolean) &lt;/li&gt;  &lt;li&gt;Prepare the {@link InterestRestrictionRequest} using the event derived in 1.1 and {@link InterestPlanItem#getInterestAgeingStatus()}&lt;/li&gt;  &lt;li&gt;Set {@link InterestRestrictionRequest#getTransactionType()} as {@link TransactionType#Credit} if {@link InterestAdjustmentRequestDTO#getEntryType()} is equal to {@link AdjustmentAmountType#INCREASE}, else set it to {@link TransactionType#Debit}. &lt;li&gt;Call  {@link InterestRestrictionAdapter#isServiceRestricted(InterestRestrictionRequest)} and receive the response isServiceRestricted&lt;/li&gt; &lt;li&gt;If isServiceRestricted == true , throw business exception with error code :  {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED}&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;Else if Product Class is {@link ProductClass#TERM_DEPOSITS} &lt;ol&gt; &lt;li&gt;Call  {@link InterestRestrictionAdapter#isAdjustmentAllowedForDeposit(String, int)} and receive the response isAllowed.&lt;/li&gt; &lt;li&gt;If isAllowed == false, throw business exception with error code : {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED_FOR_DEPOSIT}&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt;  &lt;/ol&gt; @param interestAdjustmentRequestDTO @param interestPlanItem @param accountId @param depositId @param interestRule</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	validateUserPrivilege	This method validates transaction privileges for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method is used to determine whether user has access to a particular service. @param serviceInputs - SessionContext object @throws FatalException - Exception being returned
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	validateUserPrivilege	This method validates transaction privileges for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method validates user privileges for current transaction for Perform action. @param serviceInputValues - Service details hashmap for current transaction @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	validateUserPrivilege	This method validates transaction privileges for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method validates user privileges for current transaction for PerformWithoutApprovals action. @param serviceInputs - Service details hashmap for current transaction @return Object - The List of boolean values @throws FatalException

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	validateAccessToReport	This method validates whether user has rights to access the report. @param sessionContext . @param report . @throws fatalException	This method validates whether user has rights to access the report. @param sessionContext . @param report . @throws fatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	hashProtectedResponseAttributes	This method retrieves the denied response elements. @param sessionContext @param responseObject @return Object @throws FatalException	This method hides certain values in the response objects, based on access policies defined. @param serviceInputs - SessionContext object @param responseObject - Response object, before applying security policies @throws FatalException - Exception being returned
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	hashProtectedResponseAttributes	This method retrieves the denied response elements. @param sessionContext @param responseObject @return Object @throws FatalException	This method retrieves the denied response elements. @param sessionContext @param responseObject @return Object @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	assertFraudPolicies	This method invokes the OAAM policies for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method delegates the calls to the appropriate methods during security access check process.  First of all, if the service is blacked out for the period in which the user is accessing it, then he  will not be allowed to perform the transaction.    Access policies are maintained on OID server corresponding to roles.  Based on the policies, the user  has access on a particular service or not is validated.    If the service under execution is eligible for OAAM validations, call is made to OAAM.  OAAM validates the policies and rules set for the service and returns the action to be taken.  Action can be Allow, Block, Challenge1FA, Challenge1FADelay, Challenge2FA, Challenge2FADelay.    Next assertRequiredAuthorizations is used to test for business conditions that can be either rejected,  ignored or overridden. If status is marked as overridden, system fetches the configured workflow and   distributes workitems accordingly.    @param serviceId @param context @param serviceInputs @param parameterNames @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	assertFraudPolicies	This method invokes the OAAM policies for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method is used to invoke the Policies set in OAAM. @param serviceInputs - SessionContext object @throws FatalException - Exception being returned

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	assertFraudPolicies	This method invokes the OAAM policies for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method will invoke the polices and rules set in OAAM. @param HashMap - serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap.
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	assertFraudPolicies	This method invokes the OAAM policies for user. @param serviceInputValues, This parameter captures the method arguments.  The method parameters and arguments are stored in key-value pair the HashMap. @return Object @throws fatalException , persistenceException, nestedException	This method will invoke the OAAM policies to check for Fraud Transactions and take necessary action defined in policy. It returns List of Actions which is a result of OAAM policy execution. @param,MI,HashMap,serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap. @return String - Action returned by OAAM - BLOCK, Challenge1FA, Challenge2FA, Challenge1FADelay, Challenge2FADelay @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	getUserProfileDetails	This method retrieves the user details  @fcb.param MI,String,userId, ,This represents the user ID. @return MI,UserDTO, ,The required user Information. @throws FatalException , ,Exception being returned	This method retrieves the user details  @fcb.param MI,String,userId, ,This represents the user ID. @return MI,UserDTO, ,The required user Information. @throws FatalException , ,Exception being returned
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	getUserProfileDetails	This method retrieves the user details  @fcb.param MI,String,userId, ,This represents the user ID. @return MI,UserDTO, ,The required user Information. @throws FatalException , ,Exception	This method is use to maintain the business units for the user. @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			being returned	
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	updateTransactionStatus	This method will update the transaction status in OAAM.   @param sessionId - OAAM session ID @param transactionId - OAAM transaction ID @param status - Status of OAAM transaction @param contextMap - The map of attributes required for execution of transaction. Could be null in this case. @return boolean - Status of transaction @throws FatalException	This method will update the transaction status in OAAM.   @param sessionId - OAAM session ID @param transactionId - OAAM transaction ID @param status - Status of OAAM transaction @param contextMap - The map of attributes required for execution of transaction. Could be null in this case. @return boolean - Status of transaction @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	updateTransactionStatus	This method will update the transaction status in OAAM.   @param sessionId - OAAM session ID @param transactionId - OAAM transaction ID @param status - Status of OAAM transaction @param contextMap - The map of attributes required for execution of transaction. Could be null in this case. @return boolean - Status of transaction @throws FatalException	This method will update OAAM transaction
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	updateTransactionStatus	This method will update the transaction status in OAAM.   @param sessionId - OAAM session ID @param transactionId - OAAM transaction ID @param status - Status of OAAM transaction @param contextMap - The map of attributes required for execution of transaction. Could be null in this case. @return boolean - Status of transaction @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	validateUserPrivilegePerformWOApproval		This method validates user privileges for current transaction for PerformWithoutApprovals action. @param serviceInputs - Service details hashmap for current transaction @return Object - The List of boolean values @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	initializeSecurityPolicies	This method delegates call to add annotations in classes.  The list of classes in which annotations are to be added is taken from the properties file.  @throws FatalException	This method delegates call to add annotations in classes.  The list of classes in which annotations are to be added is taken from the properties file.  @throws FatalException
IAccessPolicyAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAccessPolicyAssertionAdapter	initializeSecurityPolicies	This method delegates call to add annotations in classes.  The list of classes in which annotations are to be added is taken from the properties file.  @throws FatalException	Initialize webservice annotation processor. This injects OWSM policy on all webservices @param sessionContext
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	@param repaymentStage @param accountParameters @param definitionDate @param interestRates @param allRepaymentStages @param interestBaseHistory @param scheduleTransactionHistory
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	Will merge the original blowup with the schedule Fee Details ad create a new one @param accountId @param blowupSchedule @param scheduleFeesDetails @return SortedMap<Date, LoanRepaymentRecord> newBlowupSchedule

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	This method is used to get the due date charging data. Charging data consists of amounts to be billed with necessary billing information like interest rule code. It also has boolean flags to indicate whether given due date is interest due date and principal due date. @param account @param dueDate @return DueDateChargingData dueDateChargingData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	This method arrives at the amounts to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @return installment LoanRepaymentRecord
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	This method arrives at the principal to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param lastChargingDate @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @param interestBaseHistory @param interestRatesForStage @param accountParameters @return Money principal
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method calculates & return interest amount to schedule module @param loanEvent @param interestRates @param interestBaseHistory @param interestCalculationDateRange @param loanRepaymentStageDTO @return interestAmount	This method generates repayment records for the given stage   In EIPI Schedule, interest & fee has not been charged to user but capitalised at capitalisation dates to outstanding principal amount.  @param repaymentStage @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, LoanRepaymentRecordDTO> repaymentRecords
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getEffectiveDateForBaseChange	This method calculates & return effective Date For Base Change to schedule module @param accountId @param valueDate @param repaymentStage	This method logs principal arrear raise and payment transaction. @param scheduleId @param amount @param valueDate @param postingDate @param repaymentStage @param interestBaseHistory @param scheduleTransactionHistory @param temp TODO
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getEffectiveDateForBaseChange	This method calculates & return effective Date For Base Change to schedule module @param accountId @param valueDate @param repaymentStage	This method logs principal arrear raise and payment transaction. @param scheduleId @param amount @param valueDate @param postingDate @param repaymentStage @param interestBaseHistory @param scheduleTransactionHistory @param temp TODO
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getEffectiveDateForBaseChange	This method calculates & return effective Date For Base Change to schedule module @param accountId @param valueDate @param repaymentStage	This method logs principal payment transaction. This method is used during the rescheduling to simulate the payment of total principal arrear outstanding as of the reschedule date. @param scheduleId @param amount @param valueDate @param postingDate @param repaymentStage @param interestBaseHistory @param scheduleTransactionHistory

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getEffectiveDateForBaseChange	This method calculates & return effective Date For Base Change to schedule module @param accountId @param valueDate @param repaymentStage	This method logs partial payoff transaction @param scheduleId @param sessionId @param amount @param valueDate @param postingDate @param repaymentStage @param productCode @param interestBaseHistory @param scheduleTransactionHistory
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getEffectiveDateForBaseChange	This method calculates & return effective Date For Base Change to schedule module @param accountId @param valueDate @param repaymentStage	This method is used to Log Principal Payment while doing PPf Simulation
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getLogFromDate	This method calculates & returns Log From Date @param accountId @param onDate @return List<Object>	This method returns existing interest transactions history @param scheduleId @param onDate @param currencyCode @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getMinAdvanceInterestTerm	This method returns the minimum advance interest term. @param accountId @return	This method returns the minimum advance interest term. @param accountId @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getMinAdvanceInterestTerm	This method returns the minimum advance interest term. @param accountId @return	This method validates the temporal schedule for the given account id and session id.  @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.The information is used by the system for authentication purposes.  @fcb.param MI,String,scheduleId, , It contains the scheduleId for which the schedule is being drawn.  @fcb.param MI,Integer,sessionId, , This attribute holds the session id that is to be used by the caller  for the subsequent calls.  @fcb.param MI,Date,dateOfValidation, , This field displays the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>date of validation of schedule. &lt;br&gt; @fcb.param MI,BigDecimal,disbursementAmount, , This field displays the amount to be disbursed.&lt;br&gt; @fcb.param MI,BigDecimal,debitTypeDeductionAmount, , This field displays the total amount to be debited. &lt;br&gt; @fcb.param MI,LoanRescheduleOptionType,rescheduleOption, , This field displays option for rescheduling.Such as Recalculate term or recalculate installment.&lt;br&gt; @fcb.param MI,boolean,arrearsToBeCapitalized, , This field displays whether the arrears to be capitalized or not.&lt;br&gt; @return {@link TransactionStatus} The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Success Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.common.ErrorConstants#MID_ERECUADD}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.loan.constants.LoanErrorConstants#INVALID_SCHEDULE_SESSION}&lt;/li&gt; &lt;/ul&gt; @fcb.task LN521 @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getMinAdvanceInterestTerm	This method returns the minimum advance interest term. @param accountId @return	To check if the IO term(In case of IOA only) is in multiple of min. Advance Interest Term defined at account level at In_acct_overridable_attributes which comes from product at time of account opening. @param scheduleId @param stageNumber @param changedStageParameterDTO
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getFixedTermMonths	This method returns the fixed term months. @param accountId @return	This method returns the fixed term months. @param accountId @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getFixedTermMonths	This method returns the fixed term months. @param accountId @return	This method will check if IOA ebds before fixed rate period @param scheduleId @param newRepaymentStages
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	@param repaymentStage @param accountParameters @param definitionDate @param interestRates @param allRepaymentStages @param interestBaseHistory @param scheduleTransactionHistory
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	This method performs the reScheduleFees operation @param applicationContext @param scheduleId @param onDate @param repaymentStages @param temp TODO @param eventCode @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	Method for blown up schedule generation for new schedule and on reschedule/restructure/deferment/subsequent disbursement/top up disbursement @param accountParameters @param generationDate @param newRepaymentSchedule @param startingPrincipalBalance @param interestRates @param rescheduleOption @param scheduleTransactionHistory @param interestBaseHistory @return LoanAccountScheduleData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePreoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateFeeSchedule	@param applicationContext @param accountId @param onDate @param repaymentStages @return @throws FatalException	
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getAccountScheduleAttributes	This method returns the AccountScheduleAttributesDTO @param accountId @return	This method arrives at the amounts to be charged for the given due date and account that is in Post maturity stage @param account @param installmentDate @param stage @param interestRule @param installmentRule @return LoanRepaymentRecord repaymentRecord

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getAccountScheduleAttributes	This method returns the AccountScheduleAttributesDTO @param accountId @return	This method arrives at the amounts to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @return installment LoanRepaymentRecord
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getAccountScheduleAttributes	This method returns the AccountScheduleAttributesDTO @param accountId @return	This method arrives at the principal to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param lastChargingDate @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @param interestBaseHistory @param interestRatesForStage @param accountParameters @return Money principal
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getAccountScheduleAttributes	This method returns the AccountScheduleAttributesDTO @param accountId @return	This method calculates the number of repayments based on the given stage term, installment amount, loan present value, interest rate and repayment frequency. @param repaymentStage @param presentValue @param interestRates @paramactionDate @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	@param repaymentStage @param accountParameters @param definitionDate @param interestRates @param allRepaymentStages @param interestBaseHistory @param scheduleTransactionHistory
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	Will merge the original blowup with the schedule Fee Details ad create a new one @param accountId @param blowupSchedule @param scheduleFeesDetails @return SortedMap<Date, LoanRepaymentRecord> newBlowupSchedule

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	This method is used to get the due date charging data. Charging data consists of amounts to be billed with necessary billing information like interest rule code. It also has boolean flags to indicate whether given due date is interest due date and principal due date. @param account @param dueDate @return DueDateChargingData dueDateChargingData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	This method arrives at the amounts to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @return installment LoanRepaymentRecord
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	This method arrives at the principal to be charged for the given due date and account that is in regular stage (non-PMI stage) @param account @param lastChargingDate @param installmentDate @param stage @param interestRule @param installmentRule @param isInterestDueDate @param interestBaseHistory @param interestRatesForStage @param accountParameters @return Money principal
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getInterestAmount	This method returns the interest amount. @param applicationContext @param accountId @return	This method generates repayment records for the given stage   In EIPI Schedule, interest & fee has not been charged to user but capitalised at capitalisation dates to outstanding principal amount.  @param repaymentStage @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, LoanRepaymentRecordDTO> repaymentRecords
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getProratedFeesAtMaturityDate	This method gives prorated fees at maturity date @param applicationContext @param accountId @param eventId @param onDate @return	This method gives prorated fees at maturity date @param applicationContext @param accountId @param eventId @param onDate @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		This method does rescheduling on subsequent disbursement @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Indicates reschedule option type. Recalculate Term / Installment. @param disbursementAmount Indicates subsequent disbursement amount @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @return LoanRescheduleOutputData rescheduleOutput



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		This method does rescheduling, to be called to do reschedule when any Debit comes to loan account, caller have to pass DebitAmount along with reschedule date, this method will draw schedule on (totalOutstanding + debitAmount) and persist new schedule. After this caller have to increase principal by calling balanceService, and pass accounting entry for the same. And at the end user will update Account Utilization and then log interest computation action for Debit account in similar lines what we do for subsequent disbursement. @param applicationContext @param accountId @param rescheduleDate @param debitAmount @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		This method returns the new schedule on account deferment or deferment by interest freeze @param oldRepaymentStages @param accountParameters @param defermentDate @param splitDate @param isDefermentByInterestFreeze @param interestFreezeRateValue @param installmentRuleCodeForDeferment @param installmentTypeForDeferment @param stageDescription @param defermentTerm @param endOfPeriodTreatment @param interestRates @param interestBaseHistory @param scheduleTransactionHistory @return PrincipalScheduleDefermentOutputData deferment output
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		Common method for blown up schedule generation for new schedule and on reschedule/restructure/deferment/subsequent disbursement/top up disbursement @param sessionContext @param scheduleId @param sessionId @param rescheduleDate @param rescheduleOption @return LoanBlownupScheduleResponse @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRates		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method does rescheduling on rate change. If the input interestRates is null then fetch the rates, else use the input interest Rates. This method also saves the blown schedule and summary schedule. @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Reschedule option type. Recalculate Term / Installment. @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @param loanFeeSchedule @param interestRates @param accountEventLogType TODO @return LoanRescheduleOutputData rescheduleOutput
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method does rescheduling on subsequent disbursement @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Indicates reschedule option type. Recalculate Term / Installment. @param disbursementAmount Indicates subsequent disbursement amount @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @return LoanRescheduleOutputData rescheduleOutput
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method does rescheduling on partial payoff @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Reschedule option type. Recalculate Term / Installment. @return LoanRescheduleOutputData rescheduleOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method returns new schedule after doing reschedule for PPF. @param rescheduleOptionType @param effectiveDate @param accountParameters @param scheduleTransactionHistory @param interestBaseHistory @param oldRepaymentSchedule @param oldRepaymentStages @param splitDate @param loanFeeSchedule @return LoanAccountScheduleData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		Common method for blown up schedule generation for new schedule and on reschedule/restructure/deferment/subsequent disbursement/top up disbursement @param sessionContext @param scheduleId @param sessionId @param rescheduleDate @param rescheduleOption @return LoanBlownupScheduleResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method is used to generate summary schedule. This method calls LoanBackdatedRescheduleService for - A] defineBackdatedStages which uses the provided schedule configuration information like number of stages, installment rule code, year base type, frequencies of stages etc. to generate summary schedule. B] defineBackdatedSchedule which takes care of moratorium interest & installment amount. C] doBackdatedReschedule which actually reschedules the account based on reschedule option type. (For backdated reschedule, reschedule option type will always be RECALCULATE_INSTALLMENT) @param accountId @param rescheduleDate @param schedulePrincipalBalance @param rescheduleOption @param accountEventType @param capitalizeArrears @param loanRepaymentScheduleConfiguration @param accountParameters
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRates		This method is used to generate blown up schedule. This method calls LoanBackdatedRescheduleService for - A] generateBackdatedRecords which uses the backdated schedule generated by doBackdatedReschedule method to generate backdated blown up schedule. @param accountId @param sessionId @param repaymentStages @param scheduleInterestBase @param scheduleTransaction @param rescheduleDate @param rescheduleOption @param schedulePrincipalBalance @param accountEventType @param accountParameters

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		This method returns the simulated Periodic Prices during recommendation, when accountId is not available.And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulationForBundle		This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulationForBundle		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulationForBundle		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulation		This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulation		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulation		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getDummyPeriodicPricesSimulation		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePreoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	isRepricingRequired		This method does rescheduling on subsequent disbursement @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Indicates reschedule option type. Recalculate Term / Installment. @param disbursementAmount Indicates subsequent disbursement amount @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @return LoanRescheduleOutputData rescheduleOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	isRepricing Required		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	isRepricing Required		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	persistRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRatesMergedWithExisting		This method does rescheduling on subsequent disbursement @param applicationContext Application context @param accountId Account identifier @param rescheduleDate Date of reschedule @param rescheduleOptionType Indicates reschedule option type. Recalculate Term / Installment. @param disbursementAmount Indicates subsequent disbursement amount @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @return LoanRescheduleOutputData rescheduleOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRatesMergedWithExisting		This method does rescheduling, to be called to do reschedule when any Debit comes to loan account, caller have to pass DebitAmount along with reschedule date, this method will draw schedule on (totalOutstanding + debitAmount) and persist new schedule. After this caller have to increase principal by calling balanceService, and pass accounting entry for the same. And at the end user will update Account Utilization and then log interest computation action for Debit account in similar lines what we do for subsequent disbursement. @param applicationContext @param accountId @param rescheduleDate @param debitAmount @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchSimulatedRatesMergedWithExisting		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getArrearAmountToReduce		This is the common method to perform reschedule on rate change, subsequent disbursement and partial payoff. @param oldRepaymentStages @param accountParameters @param rescheduleDate @param splitDate @param rescheduleOption @param interestRates @param flgCapitalizeArrears @param interestBaseHistory @param scheduleTransactionHistory @return LoanRescheduleOutputData reschedule output data

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getArrearAmountToReduce		This is the common method to perform reschedule on rate change, subsequent disbursement and partial payoff. @param oldRepaymentStages @param accountParameters @param rescheduleDate @param splitDate @param rescheduleOption @param interestRates @param flgCapitalizeArrears @param interestBaseHistory @param scheduleTransactionHistory @return LoanRescheduleOutputData reschedule output data
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getNextDueAmount		This method does reschedule for the given repayment schedule. This method doesn't save the new schedule @param oldRepaymentSchedule Indicates existing schedule for the loan account @param accountParameters Loan account specific parameters required for schedule generation @param rescheduleDate Indicates date of reschedule @param rescheduleOption Indicates reschedule option type. Recalculate Term / Installment. @param interestRates Interest rate plans to be used for reschedule @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @param accountEventType Indicates loan account event type @return LoanRescheduleOutputData rescheduleOutput
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getNextDueAmount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getUnbilledPenaltyInterest		This method does reschedule for the given repayment schedule. This method doesn't save the new schedule @param oldRepaymentSchedule Indicates existing schedule for the loan account @param accountParameters Loan account specific parameters required for schedule generation @param rescheduleDate Indicates date of reschedule @param rescheduleOption Indicates reschedule option type. Recalculate Term / Installment. @param interestRates Interest rate plans to be used for reschedule @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @param accountEventType Indicates loan account event type @return LoanRescheduleOutputData rescheduleOutput
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getUnbilledPenaltyInterest		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getUnbilledPenaltyInterest		This method does reschedule for the given repayment schedule. This method doesn't save the new schedule @param oldRepaymentSchedule Indicates existing schedule for the loan account @param accountParameters Loan account specific parameters required for schedule generation @param rescheduleDate Indicates date of reschedule @param rescheduleOption Indicates reschedule option type. Recalculate Term / Installment. @param interestRates Interest rate plans to be used for reschedule @param capitalizeArrears Indicates whether unpaid arrears are to be capitalized @param accountEventType Indicates loan account event type @return LoanRescheduleOutputData rescheduleOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getUnbilledPenaltyInterest		This method calculates amount for schedule generation during restructure @param accountId @param newScheduleTypeCode @param restructureDate @param interestRates @return LoanScheduleCalculatorOutputData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getBackdatedLogFromDate	This method calculates & returns Log From Date @param accountId @param onDate @return List<Object>	This method returns existing interest transactions history @param scheduleId @param onDate @param currencyCode @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRatesAsOnDate		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRatesAsOnDate		This method is used to generate summary schedule. This method calls LoanBackdatedRescheduleService for - A] defineBackdatedStages which uses the provided schedule configuration information like number of stages, installment rule code, year base type, frequencies of stages etc. to generate summary schedule. B] defineBackdatedSchedule which takes care of moratorium interest & installment amount. C] doBackdatedReschedule which actually reschedules the account based on reschedule option type. (For backdated reschedule, reschedule option type will always be RECALCULATE_INSTALLMENT) @param accountId @param rescheduleDate @param schedulePrincipalBalance @param rescheduleOption @param accountEventType @param capitalizeArrears @param loanRepaymentScheduleConfiguration @param accountParameters

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchRatesAsOnDate		This method is used to generate blown up schedule. This method calls LoanBackdatedRescheduleService for - A] generateBackdatedRecords which uses the backdated schedule generated by doBackdatedReschedule method to generate backdated blown up schedule. @param accountId @param sessionId @param repaymentStages @param scheduleInterestBase @param scheduleTransaction @param rescheduleDate @param rescheduleOption @param schedulePrincipalBalance @param accountEventLogType @param accountParameters
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedRates		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedRates		This method is used to generate summary schedule. This method calls LoanBackdatedRescheduleService for - A] defineBackdatedStages which uses the provided schedule configuration information like number of stages, installment rule code, year base type, frequencies of stages etc. to generate summary schedule. B] defineBackdatedSchedule which takes care of moratorium interest & installment amount. C] doBackdatedReschedule which actually reschedules the account based on reschedule option type. (For backdated reschedule, reschedule option type will always be RECALCULATE_INSTALLMENT) @param accountId @param rescheduleDate @param schedulePrincipalBalance @param rescheduleOption @param accountEventType @param capitalizeArrears @param loanRepaymentScheduleConfiguration @param accountParameters

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedRates		This method is used to generate blown up schedule. This method calls LoanBackdatedRescheduleService for - A] generateBackdatedRecords which uses the backdated schedule generated by doBackdatedReschedule method to generate backdated blown up schedule. @param accountId @param sessionId @param repaymentStages @param scheduleInterestBase @param scheduleTransaction @param rescheduleDate @param rescheduleOption @param schedulePrincipalBalance @param accountEventLogType @param accountParameters
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedFees		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedFees		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedFees		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateInsurancePremiumRepayments		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateInsurancePremiumRepayments		This method will generate and save insurance premium schedule @param applicationContext @param accountParameters @param newRepaymentSchedule @param startingStageIndex @param onDate @return @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getFeeRelatedAttributes	This method return the fee related attributes. @param accountId	@param accountParameters @param scheduleEndDate @param presentValue @param totalIntInstallment @param accountId @param repaymentStages @return
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method returns the date, that is the first date after the stage start date, on which the given type of schedule event will occur. @param repaymentStage @param eventType @return Date
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method returns the date, that is the first date after the stage start date, on which the given type of schedule event will occur. @param repaymentStage @param eventType @return Date
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method Generates default schedule based on the given re payment schedule stages and account parameters but it does not divide principal amount and loan term. @param repaymentStages @param definitionDate @param amountForScheduleGeneration @param accountParameters @param interestRates

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method calculates the interest amount. This is used during schedule generation. @param accountParameters @param interestBaseHistory @param interestCalculationDateRange @param interestRates @param loanRepaymentStage @return Money @throws FatalException
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method generates repayment records for the given stage   In EIPI Schedule, interest & fee has not been charged to user but capitalised at capitalisation dates to outstanding principal amount.  @param repaymentStage @param accountParameters @param generationDate @param interestRates @param allRepaymentStages @param repaymentDates @param allRepaymentRecords @param interestBaseHistory @param scheduleTransactionHistory @return SortedMap<Date, LoanRepaymentRecordDTO> repaymentRecords

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getTransactionAttribute		This method validates the stage if one of its parameter is changed. On successful validation, it also resets other stage attributes which are affected due to change in the given stage parameter @param changedStage @param accountParameters @param onDate @param futureValue @param interestRates @param parameterType @param interestBaseHistory
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	fetchBackdatedSimulatedFeesForRollover		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getCapitalizedArrearFromBalanceHistory	This method returns the capitalized arrear amount from latest history record. @param scheduleId @param valueDate @return	This method redefines the given repayment stages having stage number equal to or greater than the given redrawStartIndex  Method calls respective stage calculators, for each stage in given schedule type, to redraw stage  @param repaymentSchedule Repayment stages @param accountParameters Loan account specific parameters required for schedule generation @param rescheduleDate Date of redraw @param redrawStartIndex All the repayment stages having stage number equal to or greater than redrawStartIndex are redefined @param interestRates Interest rate plans @param rescheduleOption Reschedule option type @param interestBaseHistory List of interest base records @param scheduleTransactionHistory List of scheduled transactions @return LoanScheduleCalculatorOutputData redrawOutput

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	computeDateForFetchingRates	This method computes the date for fetching rates. @param applicationContext @param scheduleId @param fromDate @param toDate @param scheduleSessionId @param eventList @return	This method is used to generate summary schedule. This method calls LoanBackdatedRescheduleService for - A] defineBackdatedStages which uses the provided schedule configuration information like number of stages, installment rule code, year base type, frequencies of stages etc. to generate summary schedule. B] defineBackdatedSchedule which takes care of moratorium interest & installment amount. C] doBackdatedReschedule which actually reschedules the account based on reschedule option type. (For backdated reschedule, reschedule option type will always be RECALCULATE_INSTALLMENT) @param accountId @param rescheduleDate @param schedulePrincipalBalance @param rescheduleOption @param accountEventType @param capitalizeArrears @param loanRepaymentScheduleConfiguration @param accountParameters
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	computeDateForFetchingRates	This method computes the date for fetching rates. @param applicationContext @param scheduleId @param fromDate @param toDate @param scheduleSessionId @param eventList @return	This method is used to generate blown up schedule. This method calls LoanBackdatedRescheduleService for - A] generateBackdatedRecords which uses the backdated schedule generated by doBackdatedReschedule method to generate backdated blown up schedule. @param accountId @param sessionId @param repaymentStages @param scheduleInterestBase @param scheduleTransaction @param rescheduleDate @param rescheduleOption @param schedulePrincipalBalance @param accountEventLogType @param accountParameters
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	generateNextHistoryNo		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getReview Amount		Common method for all types of reschedules i.e rate change, subsequent disbursement and partial payoff. This method generates new schedule based on the given reschedule option, generates blown up schedule and overwrites the existing schedule with newly generated one @param applicationContext @param accountParameters @param oldRepaymentSchedule @param subsequentDisbursementAmount @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param interestRates @param reasonCode @param reasonDescription @param accountEventType @return LoanRescheduleOutputData
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getReview Amount		
IScheduleLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleLoanAdapter	getReview Amount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	validateUserRoleAccessforRoles	<p>This method validates the resource access rights for a specified resource and the role. Application roles of each role specified is substituted in the subject and access checks are performed for the resource name specified. It is expected that the resource name shall be a service. The nameValuePair array represent constraints, if any, that need to be checked on the policy. If the service has been restricted for specified role and resource than the financial transaction will not be allowed to be performed in self service channel. If the service has been restricted for specified role and resource than the authorized signatory panel will not show the party signatures for assisted channel. @param templateRoleId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	validateUserAccessforRoles	<p>This method validates the resource access rights for a specified resource and the role. Application roles of each role specified is substituted in the subject and access checks are performed for the resource name specified. It is expected that the resource name shall be a service. The nameValuePair array represent constraints, if any, that need to be checked on the policy. If the service has been restricted for specified role and resource than the financial transaction will not be allowed to be performed in self service channel. If the service has been restricted for specified role and resource than the authorized signatory panel will not show the party signatures for assisted channel. @param templateRoleId</p>	<p>&lt;p&gt;This method check whether the facility for given party is valid.This method works as follows: Check if the&lt;br /&gt; Channel is Branch,then the party in party account relationship has Branch signatory flag set. if the Channel is&lt;br /&gt; Internet,then the party in party account relationship should have Internet signatory flag set. if the Channel is&lt;br /&gt; Mobile Banking,then the party in party account relationship should have Mobile Banking signatory flag set. Only&lt;br /&gt; if the flags are set then the party would be added to the list of parties who can operate on the account. &lt;/p&gt;         @param sessionContext @param iAccountPartyAdapter @param operatingInstructionPartyAccountRelationshipDetails DTO @param count @param partyAccountRelationshipDetailsDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	validateUserAccessforRoles	This method validates the resource access rights for a specified resource and the role. Application roles of each role specified is substituted in the subject and access checks are performed for the resource name specified. It is expected that the resource name shall be a service. The nameValuePair array represent constraints, if any, that need to be checked on the policy. If the service has been restricted for specified role and resource than the financial transaction will not be allowed to be performed in self service channel. If the service has been restricted for specified role and resource than the authorized signatory panel will not show the party signatures for assisted channel. @param templateRoleId	<p>This method would prepare the OperatingInstructionPartyDetailsDTO.It works as follows: ;For each  OperatingInstructionPartyDetailsDTO, if the operating instruction channel is Branch: Check the Branch signatory  flag set in party account relationship. Check if the KYC for the party is confirmed. Check if the party has valid   user access role(template/account role) defined in OES for the given service. If all the above is true,the party  would be added to list of parties who can operate on the account.If the operating  instruction channel is Internet: Check the Internet signatory flag set in party account relationship Check if the   KYC for the party is confirmed Check if the party has valid user access role(template/account role) defined in  OES for the given service. If all the above is true,the party would be added to list of parties who can operate  on the account.If the operating instruction channel is Mobile Banking: Check the Mobile  Banking signatory flag set in party account relationship Check if the KYC for the party is confirmed Check if the   party has valid user access role(template/account role) defined in OES for the given service. If all the above is   true,the party would be added to list of parties who can operate on the account.; </p> @param sessionContext @param returnDTO @param operatingInstructionPartyDetailsDTOs @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	validateUserAccessforRoles	This method validates the resource access rights for a specified resource and the role. Application roles of each role specified is substituted in the subject and access checks are performed for the resource name specified. It is expected that the resource name shall be a service. The nameValuePairs array represent constraints, if any, that need to be checked on the policy. If the service has been restricted for specified role and resource than the financial transaction will not be allowed to be performed in self service channel. If the service has been restricted for specified role and resource than the authorized signatory panel will not show the party signatures for assisted channel. @param templateRoleId	This method validates the resource access rights for a specified resource and the role. Application roles of each role specified is substituted in the subject and access checks are performed for the resource name specified. It is expected that the resource name shall be a service. The nameValuePairs array represent constraints, if any, that need to be checked on the policy. If the service has been restricted for specified role and resource than the financial transaction will not be allowed to be performed in self service channel. If the service has been restricted for specified role and resource than the authorized signatory panel will not show the party signatures for assisted channel. @param templateRoleId
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	isAccountValid		This method used to load the loan account based on account number. @param accountId @return account-Details of the loan account.
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSMS Adapter	isAccountValid		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		This method checks the validity of the specified account number.   It is invoked by every maintenance method of this service to check whether the specified account is valid or whether it exists or not.  @param sessionContext @param accountId @param accountType @throws FatalException
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		Method will load {@link DemandDepositAccount} with given accountId. @param accountId @return DemandDepositAccount
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		This method is used to fetch all the accounts which open on today's posting date. @param accountId @return DemandDepositAccount[]
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		This method used to load the Term deposit account. @param accountId @return account-Details of the TD account.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		
IAccountSMS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSMSAdapter	isAccountValid		
IFeeNegotiationApprovalPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IFeeNegotiationApprovalPricingAdapter	updatePriceVariance	This method authorizes the price variance for the given unique udm code @param sessionContext @param pricingTicketNumber @throws FatalException	
IFeeNegotiationApprovalPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IFeeNegotiationApprovalPricingAdapter	updatePriceVariance	This method authorizes the price variance for the given unique udm code @param sessionContext @param pricingTicketNumber @throws FatalException	
IFeeNegotiationApprovalPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\IFeeNegotiationApprovalPricingAdapter	updateUDMApprovalStatus		
ICommonServiceBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons\ICommonServiceBropAdapter	fetchBySearchCriteria	This method will fetch batch operations for a given search criteria. Batch type is a mandatory parameter. @param batchOperationSearchDTO @return BatchStatusInquiryResponse @throws FatalException	Validation method to check if branch batch is open. @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBropAdapter	fetchBySearchCriteria	This method will fetch batch operations for a given search criteria. Batch type is a mandatory parameter. @param batchOperationSearchDTO @return BatchStatusInquiryResponse @throws FatalException	Validation method to check if branch batch is open. @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @throws com.ofss.fc.infra.exception.FatalException
ICommonServiceBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBropAdapter	fetchBySearchCriteria	This method will fetch batch operations for a given search criteria. Batch type is a mandatory parameter. @param batchOperationSearchDTO @return BatchStatusInquiryResponse @throws FatalException	This method will fetch batch operations for a given search criteria. Batch type is a mandatory parameter. @param batchOperationSearchDTO @return BatchStatusInquiryResponse @throws FatalException
ICommonServiceBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBropAdapter	addBranchDetails	This method will be called at the time of branch creation in Common Services module, to populate the details of newly created branch in BROP module	This represents the addCldrWeeklyHolidayWork method of the entity. @param MI MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI MI,CldrWeeklyHolidayWorkDTO,cldrWeeklyHolidayWorkDTO, ,cldrWeeklyHolidayWorkDTO TODO: Explain the cldrWeeklyHolidayWorkDTO above @return {@link TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILegalProcessesForApprovalCallingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ILegalProcessesForApprovalCallingAdapter	callApproval	@param legalProcessForApprovalRequestDTO @return TransactionStatus @throws FatalException	@param LegalProcessForApprovalRequestDTO @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		<p>This method will check that the nominated account for the bundle must be either a CASA or a LOAN account. </p> @param partyBundleLinkage @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		This method fetches product class for the input product code. @param sessionContext @param productCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		This method performs all the validations required for CRUD operations. In case of the mode being ADDITION or MODIFICATION, following methods are called <li>checkMandatoryAttributes()</li> <li>validateAccount()</li> <li>fetchAndValidateProductClass()</li> <li>validateForMaximumNominationsLimit()</li> <li>validateAllParties()</li> In case of mode being INQUIRE or DELETION, following methods are called <li>checkMandatoryAttributes()</li> <li>validateAccount()</li> @param sessionContext @param accountNominationRequestDTO @param mode @throws FatalException @throws BusinessException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClass		<p>This method will be used to fetch the next charging date for bundle maintenance fee. When bundle subscription is done online, the computation date for periodic fee will be available and {@linkplain com.ofss.fc.app.pricing.service.contract.EventPriceApplicationService.getPeriodicPricesWithPeriodicDates} service will be called to fetch the next charging date. However, when bundle subscription request is made via Origination, the computation date is not available and {@linkplain com.ofss.fc.app.pricing.service.contract.PeriodicFeeSimulationApplicationService.computePeriodicFeeDates} service will be called to fetch the next charging date.</p> @param partyBundleLinkage
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		<p>This method will fetch the product group id given the product id of the nominated account. </p> @param sessionContext @param productId @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		This method provides information required by origination module.Information contains. <ol> <li>Basic OD account details. <li>Worst current DPD(days passed due) : Oldest arrear. <li>No. Of cheque returned: Cheque returned as part of inward clearing in last 12 months. </ol> If effective is null, posting date will be considered for calculation. @param accountId @param effectiveDate @return transactionAccountConductDetailsDTO @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroup		This method is used to fetched allowed relief types for populating the matrix. @param sessionContext @param accountId @param hardshipReliefTypes @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		<p>This method is responsible for placing the Request for Loan Statement Generation. For the first time, The Statement generation request has been placed. This method is invoked during Loan Account Opening process. </p> <p>@param loanAccountOpeningDTO @return</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		<p>This method returns statement parameters details for the given productCode. The accountId is required only for logging. </p> <p>@param productCode @return</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		<p>This method takes care of all the consolidation details that is required and returns the inquiry response object filled with the statement items inquired. It follows the following steps:</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Initialize sortedDDStatementItemDTOSet:TreeSet&lt;DDStatementItemDTO&gt; using the postingDateComparator&lt;/li&gt;</li> <li>&lt;li&gt;Initialize a {@link StatementStatistics} object statementStatistics, and {@link DDStatementItemConsolidator} object dDStatementItemConsolidator&lt;/li&gt;</li> <li>&lt;li&gt;Fetch the isChequeConsolidationRequired and isInteresAndTaxConsolidationRequired using the account product adaptor.&lt;/li&gt;</li> <li>&lt;li&gt;Loop thorough all the financial statement items in {@link StatementInquiryResponse}. Do the following: <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch the consolidation type of the current item.&lt;/li&gt;</li> <li>&lt;li&gt;If the type equals ConsolidationType.NONE, Assemble the item, add it to the sortedDDStatementItemDTOSet and update statementStatistics. &lt;/li&gt;</li> <li>&lt;li&gt;If the type equals ConsolidationType.BULKFEE, add it to the dDStatementItemConsolidator&lt;/li&gt;</li> <li>&lt;li&gt;If the type equals ConsolidationType.CHEQUE and</li> </ul> </li> </ul> </li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>isChequeConsolidationRequired, add it to dDStatementItemConsolidator&lt;/li&gt; &lt;li&gt;If the type equals ConsolidationType.INTEREST_AND_TAX and isInteresAndTaxConsolidationRequired, add it to dDStatementItemConsolidator&lt;/li&gt; &lt;/ol&gt; &lt;li&gt;Call the {@link DDStatementItemConsolidator#fetchStatementItems} and add the result list to sortedDDStatementItemDTOSet.&lt;/li&gt; &lt;li&gt;Loop through all the all financial statement items in statementInquiryResponse and add it to sortedDDStatementItemDTOSet after assembling them.&lt;/li&gt; &lt;li&gt;Convert the TreeSet into {@link DDStatementItemDTO}[] and set it in the DDTransactionInquiryResponse along with statementStaticsics and return the same.&lt;/li&gt; &lt;/ol&gt;</p> <p>@param sessionContext @param ddTransactionInquiryResponseDTO @param statementInquiryResponse @param postingDateComparator @return {@link DDTransactionInquiryResponse} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		<p>This method takes care of all the consolidation details that is required and returns the inquiry response object filled with the statement items inquired. It follows the following steps: <ol style="list-style-type: none"> <li>Initialize sortedTDStatementItemDTOSet:TreeSet&lt;TDStatementItemDTO&gt; using the postingDateComparator</li> <li>Initialize a {@link StatementStatistics} object statementStatistics, and {@link TDStatementItemConsolidator} object tDStatementItemConsolidator</li> <li>Fetch the isChequeConsolidationRequired and isInteresAndTaxConsolidationRequired using the account product adaptor.</li> <li>Loop thorough all the financial statement items in {@link StatementInquiryResponse}. Do the following: <ol style="list-style-type: none"> <li>Fetch the consolidation type of the current item.</li> <li>If the type equals ConsolidationType.NONE, Assemble the item, add it to the sortedDDStatementItemDTOSet and update statementStatistics.</li> <li>If the type equals ConsolidationType.BULKFEE, add it to the dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.CHEQUE and isChequeConsolidationRequired, add it to dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.INTEREST_AND_TAX and isInteresAndTaxConsolidationRequired, add it to dDStatementItemConsolidator</li> </ol></li> <li>Call the {@link TDStatementItemConsolidator#fetchStatementItems} and add the result list to sortedTDStatementItemDTOSet.</li> <li>Loop through all the all financial statement items in statementInquiryResponse and add it to sortedDDStatementItemDTOSet after assembling</li> </ol></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>them.</p> <p>Convert the TreeSet into <a href="#">TStatementItemDTO</a> and set it in the DDTransactionInquiryResponse along with statementStaticsics and return the same.</p> <p>@param sessionContext @param tdTransactionInquiryResponseDTO @param statementInquiryResponse @param postingDateComparator @return TDTransactionInquiryResponse @throws FatalException</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method is used to populate the DTO for logging the alert on Adhoc Statement Generation on Printing @param accountId @param object @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		<p>This method is used to inquire preferences set for demand deposit account.Account can have following preferences</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Facilities : &lt;ol&gt; &lt;li&gt;Common facilities: instructions( standing ,sweep out),temporary overdraft,underline. Please refer <a href="#">{@link AccountFacilities}</a>. &lt;li&gt;External banking facilities : Facilities which allows transactions from different channels. Please refer <a href="#">{@link BankingFacilities}</a>. &lt;/ol&gt;</li> <li>&lt;li&gt;Statement preference : Account statement setup parameters which include online statement,physical statement,statement delivery options.Please refer <a href="#">{@link StatementPreference}</a> &lt;li&gt;Cheque book preferences : If cheque book facility is available, some of the parameters can be overridden. Cheque book parameters such as reorder level ,number of cheques etc.Please refer <a href="#">{@link ChequeBookPreferences}</a> for more details. &lt;li&gt;Minimum balance preferences : Define minimum balance required for given period of time. Please refer <a href="#">{@link MinimumBalancePreferences}</a>. &lt;li&gt;Common preferences : such as relationship officer,employee account etc. &lt;/ol&gt;</li> </ul> <p>This method performs following steps to get the preferences.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Load demand deposit account: Call <a href="#">{@link DemandDepositAccountService#fetchDemandDepositAccount}</a> &lt;li&gt;Call assembler to get <a href="#">{@link DDAPreferencesDTO}</a> : Call <a href="#">{@link DDAPreferencesAssembler#fromDemandDepositAccount}</a> &lt;li&gt;Prepare response object using DTO created in second step. &lt;/ol&gt;</li> </ul> <p><a href="#">@fcb.param MI,SessionContext,sessionContext, ,</a> This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. <a href="#">@fcb.param MI,String,accountId,</a> &lt;br&gt; The attribute required to</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>inquire preferences associated with it. @return {@link DDAPreferencesInquiryResponse} &lt;br&gt;  DDAPreferencesInquiryResponse contains DDAPreferencesDTO which will contain following information.&lt;br&gt;  &lt;b&gt;accountId&lt;/b&gt;DemandDepositAccountId.&lt;br&gt;  &lt;b&gt;StatementPreferencesDTO&lt;/b&gt; Statement preferences which will contain following information&lt;br&gt; - statementDeliveryMode&lt;br&gt; - physicalStatementFrequency&lt;br&gt; - onlineStatementFrequency&lt;br&gt; - statementToAllHoldersFlag&lt;br&gt; &lt;b&gt;ChequeBookPreferencesDTO&lt;/b&gt; CheckBook preferences which will contain following information&lt;br&gt; - reorderLevelPercentageForLocal&lt;br&gt; - reorderLevelPercentageForMulticity&lt;br&gt; - maxNoOfLeavesForLocal&lt;br&gt; - maxNoOfLeavesForMulticity&lt;br&gt; &lt;b&gt;MinimumBalancePreferencesDTO&lt;/b&gt; MinimumBalance preferences which will contain following information&lt;br&gt; - minimumBalanceRequired&lt;br&gt; - validityStartDate&lt;br&gt; - validityEndDate&lt;br&gt; &lt;b&gt;AccountFacilitiesDTO&lt;/b&gt; AccountFacilities which will contain following information&lt;br&gt; - hasChequeBook&lt;br&gt; - hasInwardDirectDebit&lt;br&gt; - hasStandingInstruction&lt;br&gt; - hasSweepinInstruction&lt;br&gt; - hasSweepoutInstruction&lt;br&gt; - hasOverline&lt;br&gt; - hasTOD&lt;br&gt; &lt;b&gt;BankingFacilitiesDTO&lt;/b&gt; BankingFacilities which will contain following information&lt;br&gt; - hasATM&lt;br&gt; - hasInternetBanking&lt;br&gt; - hasMobileBanking&lt;br&gt; - hasTeleBanking&lt;br&gt; - hasPOS&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>hasDebitCard&lt;br&gt; &lt;b&gt;AccountPreferenceDTO&lt;/b&gt;  AccountPreference which will contain following information&lt;br&gt; -isMailToBeHeld&lt;br&gt; -isAccountRestricted&lt;br&gt; -isEmployeeAccount&lt;br&gt; -mailAddressControl&lt;br&gt; -minorAccountStatus&lt;br&gt; -relationshipOfficerId&lt;br&gt; In case of failure -&lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; &lt;ul&gt; &lt;li&gt; {@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		@param productCode @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		This method returns statement parameters details for the given productCode. The accountId is required only for logging. @param productCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementParameters		Method will be used to validate all the facilities {@link BankingFacilities} and {@link AccountFacilities} provided to account with Product defined under.
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEventDetails	@param eventId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEventDetails	@param eventId @return	This method would call the product service to fetch the domain category for a given event id, which will be used in raising accounting in case of non-domain category like CH,AT.
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEventDetails	@param eventId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEventDetails	@param eventId @return	This method will populate the event description based on the event id given. If the event id is given as null, it will Throw a business exception. @param eventId @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEventDetails	@param eventId @return	<p>This method is to raise accounting for all the fees applicable on the event. As per the bank level parameter (show discount) will include the discount accounting .</p> <p>Following are the required steps:</p> <ol style="list-style-type: none"> <li>1. populate accounting event DTO {<a href="#">@link com.ofss.fc.app.accounting.dto.AccountingEventDTO</a>}</li> <li>2. Get instance of Account accounting adapter {<a href="#">@link com.ofss.fc.app.accounting.dto.AccountingEventDTO</a>}</li> <li>3. Call raise Accounting of adapter. {<a href="#">@link com.ofss.fc.app.adapter.impl.account.AccountAccountingAdapter#raiseAccountingEvent</a>}</li> </ol> <p>@param primaryAccountId @param accountingEventDetailDTO @param accrualStatusType @param taskModuleType @param eventCode @param transactionDescription @param transactionRefNo @param accountBranch @param productCode @param partyId @throws FatalException</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	getNonProductDomainCategory	@param eventCode @return	<p>This method is responsible for levying Periodic Fees on the account in case bulking flag is true.</p> <p>Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.</p> <p>The bulking indicator and the frequency is setup in pricing module.</p> <p>Once the fee details are populated by the caller method this method internally gives a call to levyGiven which in turn will debit the account and intimate pricing.</p> <p>@param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	getNonProductDomainCategory	@param eventCode @return	This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	getNonProductDomainCategory	@param eventCode @return	This method will fetch the non product domain category @param eventCode @return NonProductDomainCategory
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	getNonProductDomainCategory	@param eventCode @return	This method is used to raise the accounting event for the transfer of refund of fees @param accountId : Indicates the parent in case the refund is done on alternate account, otherwise it is the self. @param accountingEventAmountTagDetailsDTOs @param reversalDetailDTO @param applicationContext
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchAmountTagForPriceDefintion	@param pdName @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchAmountTagForPriceDefintion	@param pdName @return	Build the credit leg for accounting event @param pdCode @param domainCategory @param eventCode @param netAmount @param valueDate @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchAmountTagForPriceDefintion	@param pdName @return	Method to fetch mountTags from PMU For PDID @param componentPreferenceDTO @param feeAmount @param eventCode @return PriceDefinitionAmountTagDTO[]

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductClassCurrent	Check the product class from pmu module @param productId @return	This method returns true or false depending on the product class from product manufacturing account is current or savings @param productCode @return @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductClassCurrent	Check the product class from pmu module @param productId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClassFromPMU		This method fetches product class for the input product code. @param sessionContext @param productCode @return @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isBundleIdValid	This method takes the bundle id as input and checks if the bundle id is valid. @param bundleId @return @throws FatalException	This method will check if the bundle id supplied for bundle subscription is valid. @param partyBundleLinkageBusinessPolicyDTO @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isBundleIdValid	This method takes the bundle id as input and checks if the bundle id is valid. @param bundleId @return @throws FatalException	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will be called when a request either to link or delink and account to a bundle is made. </p> @param newBundleAccountLinkageArray @param newAccountBasicDetailsDTOArray @param partyBundleLinkage @param isLink @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will make an adapter call to PMU and fetch all offers belonging to the given bundle id. </p> @param bundleId @return @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will validate if bundle status changed on charging subscription fee. </p> @param partyBundleLinkage @param isBundleBenefitApplicable @param existingBundleAccountLinkageList @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method is used when a single or multiple parties subscribe for a bundle. It returns the bundle reference id once the party bundle linkage is created. </p> @param partyBundleLinkage @return string

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method is used when the user attempts to add or delete owners to an already subscribed bundle. </p> @param partyBundleLinkage @param accountBundleLinkage
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will fetch offer details under the bundle id for the input bundle reference id. </p> @param bundleReferenceId @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will be used to validate bundle subscription.  The {@link com.ofss.fc.enumeration.ServiceCallContextType} ServiceCallContextType will be set to VALIDATE here.  If accounts have been provided to be linked to the bundle at the time of subscription, this method will additionally validate the link account to bundle business policy. </p> @fcb.param MI, ApplicationContext, applicationContext, Standard header information for the application. @fcb.param MI, BundleSubscriptionDTO, bundleSubscriptionDTO, This dto holds the request details for bundle subscription action. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method reads the bank policy for bundle owners and validates if the input list of bundle owners with the chosen offers, are eligible for bundle subscription. </p> @fcb.param MI, ApplicationContext, applicationContext, Standard header information for the application. @fcb.param MI, BundleSubscriptionDTO, bundleSubscriptionDTO, This dto holds the request details for bundle subscription action. @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will charge the bundle subscription fee from the nominated account and raise corresponding alert. If the ChargeActionType is chosen as COLLECT, this method will debit the account, raise accounting event, intimate pricing and raise alert. If the ChargeActionType is choosen as WAIVE, this method will only intimate pricing and raise alert. </p> @fcb.param MI, ApplicationContext, applicationContext, Standard header information for the application. @fcb.param MI, AccountDetailsDTO, accountDetailsDTO, Holds basic details to be linked to the bundle during subscription. @fcb.param MI, IPartyBundleLinkage, partyBundleLinkage, Indicates subscription details. @fcb.param MI, AccountBasicDetailsDTO[], accountBasicDetailsDTOArray, Indicates details of accounts provided for subscription. @fcb.param MI, String, bundleReferenceId, Indicates the unique identifier for party bundle linkage. @fcb.param MI, String, pricingTicketNumber, Indicates the ticket code against which the bundle subscription fee is to be charged. @fcb.param MI,, WrappedPriceFeatureDTO, wrappedPriceFeatureDTO , Holds details of the bundle subscription fee. @return Money, feeCharged, Indicates the amount charged as bundle subscription fee. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will return all applicants as the bundle owner when bank policy for bundle owner strategy reads ALL MANDATORY. </p> @fcb.param MI, SubscriptionDetailsDTO, subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String, bundleId, Indicates the bundle chosen for subscription. @fcb.param BundleOfferDTO, bundleOfferDTO, Indicates the details of the input bundle id. @return Set<String>, Set of identified bundle owners.
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	<p>This method will validate that all the offers chosen for subscription must have been applied by identified bundle owners. </p> @param partyIdArray @param subscriptionDetailsDTOArray @param bundleId
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleDetails	This method takes the bundle id as input and fetches the details of the offers belonging to this bundle. Details include the offer id and whether the offer is mandatory or optional under the bundle. @param bundleId @return @throws FatalException	This method will make an adapter call to PMU and fetch all bundles which are available for subscription i.e. all bundles where subscription date is less than equal to the current posting date.   @param valueDate
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	This service will return the set of all eligible offset offer. @param accountId @param currencyCode @return Set<String>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountConfigurationDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEligibleOffersForLinkage	Fetch the eligible offset account offers IDs and currency given the beneficiary account offer id and beneficiary account offer currency. @param sessionContext @param offerId @param offerCcy @return	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroupDetails		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroupDetails		This method provides information required by origination module. Information contains. <ol style="list-style-type: none"><li>Basic OD account details.</li><li>Worst current DPD(days passed due) : Oldest arrear.</li><li>No. Of cheque returned: Cheque returned as part of inward clearing in last 12 months.</li></ol> If effective is null, posting date will be considered for calculation. @param accountId @param effectiveDate @return transactionAccountConductDetailsDTO @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductGroupDetails		This method is used to fetched allowed relief types for populating the matrix. @param sessionContext @param accountId @param hardshipReliefTypes @return @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClassForOffer	<p>This method will fetch product class, given the offer id. </p> @param offerId @return @throws FatalException	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClassForOffer	<p>This method will fetch product class, given the offer id. </p> @param offerId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClassForOffer	<p>This method will fetch product class, given the offer id. </p> @param offerId @return @throws FatalException	<p>This method will check that the nominated account for the bundle must be either a CASA or a LOAN account. </p> @param partyBundleLinkage @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchProductClassForOffer	<p>This method will fetch product class, given the offer id. </p> @param offerId @return @throws FatalException	<p>This method will be used to fetch the next charging date for bundle maintenance fee. When bundle subscription is done online, the computation date for periodic fee will be available and {@linkplain com.ofss.fc.app.pricing.service.contract.EventPriceApplicationService.getPeriodicPricesWithPeriodicDates} service will be called to fetch the next charging date. However, when bundle subscription request is made via Origination, the computation date is not available and {@linkplain com.ofss.fc.app.pricing.service.contract.PeriodicFeeSimulationApplicationService.computePeriodicFeeDates} service will be called to fetch the next charging date. </p> @param partyBundleLinkage
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLedgerCodeForDomainCategory		
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLedgerCodeForDomainCategory		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLedgerCodeForDo mainCategory		This method executes the settlement payout instruction and raises an accounting event for the same.   It does the following,    <ol> <li>It loads {@link SettlementPayoutItem} via {@link SettlementPayoutInstructionItemsRepository} to obtain settlement reference number (i.e componentPreferenceId)</li> <li>Executes Settlement Instruction via {@link IAccountSettlementAdapter#executePreference} method.</li> <li>If execution of settlement instruction is successful (settlementExecutionResponse.getSummary().getUnsettledAmount().isZero()), it raises an accounting event via {@link SettlementInstructionPayoutApplication#raiseAccounting} method else thrown a {@link BusinessException}</li> </ol> @param applicationContext @param instructionId
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLedgerCodeForDo mainCategory		This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLedgerCodeForDomainCategory		This method will be called for fetching the component preference DTOs maintained for contribution, disbursement and settlement payout. Fetch method of settlement payout will internally call this method for fetching the componentPreferenceDTO maintained for all the instructions. This will then call inquireComponentPreferenceDetails() method for fetching the details for that instruction and account Id through account settlement adapter. @param sessionContext @param settlementPayoutInstructionDTO @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchStatementGenerationDateType		@param productCode @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	validateEventAllowedForOffer	To validate if the event is allowed for the offerid @param eventId @param offerId @return @throws FatalException	This method will call the offer service to check whether the event is allowed for the offer. This method will return true if the transaction is allowed. @param serviceId @param eventId @param offerId @return @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	validateEventAllowedForOffer	To validate if the event is allowed for the offerid @param eventId @param offerId @return @throws FatalException	This method will validate whether the transaction is allowed for the offer under which the account is opened. This method will be called from the validateStatus method and from the places where the explicit call is required a separate call would be made.  Call to be made to the PMU which will validate.  The flag would be sent either true or false.  If true the transaction would continue otherwise the business exception would be thrown "This transaction<Event> is not valid for the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				offer<Offerid> @param serviceId @param eventId @param offerId
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @fcb.param MI, String, eventId @return @throws FatalException	This is a private method which will be used to populate the information object AccountTransaction from the inputDTO. For events which have accounting domain as CS, TD or LN, rate type from RateSettlementMode should be inquired for product code and event and for others, it is to be inquired with non-product domain category and event. @param sessionContext @param clearingTransactionDTO @param account @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @fcb.param MI, String, eventId @return @throws FatalException	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @fcb.param MI, String, eventId @return @throws FatalException	This method populates AccountingEventDTO which is required to raise accounting event. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, TransactionLogDTO, transactionLog <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}</li> @fcb.param MI, List, accountingEventAmountTagDetailsDTO @fcb.param MI, String, eventId @fcb.param MI, MessageLogItemDTO, messageLogItem <li>{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}</li> @return PCAccountingEventDTO <li>{@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDT

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				O}</li> @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @fcb.param MI, String, eventId @return @throws FatalException	This will populate the PAccountingEventDTO which is required to raise accounting for bank draft. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, BankDraftDTO, bankDraftDTO <li>{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}</li> @fcb.param MI, String eventId @fcb.param MI, List, accountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @return pcAccountingEventDTO <li>{@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id, which would be used in raising accounting in case of non-domain category like CH/AT. @fcb.param MI, String, eventId @return @throws FatalException	This method would call the product service to fetch the domain category for a given event id, which will be used in raising accounting in case of non-domain category like CH,AT.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id, which would be used in raising accounting in case of non-domain category like CH/AT. @fcb.param MI, String, eventId @return @throws FatalException	This method populates the AccountingEventDTO in order to raise financial event for cash deposit. @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventDetailDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, CashTransactionDTO, cashTransactionDTO <li> {@link com.ofss.fc.app.dda.dto.transaction.CashTransactionDTO}</li> @return accountingEventDTO <li>{@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}</li> @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id, which would be used in raising accounting in case of non-domain category like CH/AT. @fcb.param MI, String, eventId @return @throws FatalException	This method is used call the product service to fetch domain category from the given event code. @fcb.param MI, String, eventId @return NonProductDomainCategory<li> {@link com.ofss.fc.enumeration.productmanufacturing.NonProductDomainCategory}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id, which would be used in raising accounting in case of non-domain category like CH/AT. @fcb.param MI, String, eventId @return @throws FatalException	<p>This method is to raise accounting for all the fees applicable on the event. As per the bank level parameter (show_discount) will include the discount accounting. </p> <p> Following are the required steps:  1. populate accounting event DTO {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}</p> <p> 2. Get instance of Account accounting adapter   {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}</p> <p> 3. Call raise Accounting of adapter.  {@link com.ofss.fc.app.adapter.impl.account.AccountAccountingAdapter#raiseAccountingEvent}</p> @param primaryAccountId @param accountingEventDetailDTO @param accrualStatusType @param taskModuleType @param eventCode @param transactionDescription @param transactionRefNo @param accountBranch @param productCode @param partyId @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @fcb.param MI, String, productCode @return @throws FatalException	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @fcb.param MI, FundExchangeType, fundExchangeType @fcb.param MI, CollectionType, collectionType @fcb.param MI, String, endpointId @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @fcb.param MI, FundExchangeType, fundExchangeType @fcb.param MI, CollectionType, collectionType @fcb.param MI, String, endpointId @return @throws FatalException	This method builds the debit leg. @fcb.param MI, TransactionLogDTO, transactionLogDTO <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}</li> @fcb.param MI, int, sequenceNumber @fcb.param MI, List, accountingEventAmountTagDetailsDTO @fcb.param MI, String, drAmountTag @fcb.param MI, InstrumentLogItemDTO, instrumentLogItemDTO <li>{@link com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO}</li> @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @fcb.param MI, FundExchangeType, fundExchangeType @fcb.param MI, CollectionType, collectionType @fcb.param MI, String, endpointId @return @throws FatalException	This method fetches the endpoint ledger. @fcb.param MI, TransactionLogDTO, transactionLogDTO <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}</li> @fcb.param MI, MessageLogItemDTO, messageLogItemDTO <li>{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}</li> @return endpointLedgerId @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchActiveBundles	This method fetches the active bundles. @param valueDate @return @throws FatalException	This method will make an adapter call to PMU and fetch all bundles which are available for subscription i.e. all bundles where subscription date is less than equal to the current posting date.   @param valueDate
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	verifyIfEventIsProductOrNonProduct	This method will check if the input event id has accounting domain category as product or non-product. This needs to be verified before raising accounting and the same is to be raised accordingly. @param eventId @return	This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	verifyIfEventsIsProductOrNonProduct	This method will check if the input event id has accounting domain category as product or non-product. This needs to be verified before raising accounting and the same is to be raised accordingly. @param eventId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	verifyIfEventsIsProductOrNonProduct	This method will check if the input event id has accounting domain category as product or non-product. This needs to be verified before raising accounting and the same is to be raised accordingly. @param eventId @return	<p>&lt;p&gt;This method is to raise accounting for all the fees applicable on the event. As per the bank level parameter (show discount) will include the discount accounting. &lt;/p&gt;</p> <p>&lt;p&gt;Following are the required steps:&lt;br&gt; 1. populate accounting event DTO {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt;</p> <p>&lt;p&gt; 2. Get instance of Account accounting adapter &lt;br&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt;</p> <p>&lt;p&gt; 3. Call raise Accounting of adapter.&lt;br&gt; {@link com.ofss.fc.app.adapter.impl.account.AccountAccountingAdapter#raiseAccountingEvent}&lt;/p&gt;</p> <p>@param primaryAccountId @param accountingEventDetailDTO @param accrualStatusType @param taskModuleType @param eventCode @param transactionDescription @param transactionRefNo @param accountBranch @param productCode @param partyId @throws FatalException</p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchCurrenciesForOffer	This method will fetch the eligible currencies for the offer id linked to a product.. @param eventId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchCurrenciesForOffer	This method will fetch the eligible currencies for the offer id linked to a product.. @param eventId @return	This method returns the eligible currencies for a offer. @param offerId @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDeductionModeTypeOnEvent	<p>This method will fetch the fee charging deduction mode for the input product id. This is a set up against event code in PMU and if no matching data is found, this method will return the default DeductionModeType as BILL. </p> @param productId @param eventId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDeductionModeTypeOnEvent	<p>This method will fetch the fee charging deduction mode for the input product id. This is a set up against event code in PMU and if no matching data is found, this method will return the default DeductionModeType as BILL. </p> @param productId @param eventId @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	loadBundleDetails	<p> For the input bundleId, this method will fetch all the details of the bundle. </p> @param bundleId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	loadBundleDetails	<p> For the input bundleId, this method will fetch all the details of the bundle. </p> @param bundleId @return	<p>This method will load the input bundle id. It will check the mandatory offers in the bundle and validate it against the mandatory offers chosen for subscription. </p> <p> If the caller is Origination, this method will validate that if a bundle has one or more sub-sets defined, then minimum number of offers required within each sub-set must be selected for subscription. </p> <p> If the caller is not Origination, then apart from the above, this method will also validate that all mandatory offers within the chosen bundle must be selected for subscription. </p> @fcb.param MI, ApplicationContext, applicationContext, Standard header information for the application. @fcb.param MI, {@link com.ofss.fc.app.account.offerbundling.dto.BundleOfferDTO}, bundleOfferDTO, This dto holds the selected bundle id and the offer ids chosen for subscription. @fcb.param MI, isExisting, boolean, Indicates if the bundle being subscribed is existing bundle or new one. @throws FatalException
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleBankPolicy	<p>This method will fetch the bundle bank policy definition for the input bank code. </p> @param bankCode @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleBankPolicy	<p>This method will fetch the bundle bank policy definition for the input bank code. </p> @param bankCode @return	<p>This method will validate bank policy definition for link account and bundle subscription request.  </p> @param bundleAccountLinkageList @param bundleOfferDTO @param partyBundleLinkage @param isBundleBenefitApplicable @param isBatch @param feeCharged @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleBankPolicy	<p>This method will fetch the bundle bank policy definition for the input bank code. </p> @param bankCode @return	<p>This method will validate bank policy definition for delink account request. </p> @param bundleAccountLinkageList @param bundleAccountLinkageToBeDelinkedList @param bundleOfferDTO @param partyBundleLinkage @param isBundleBenefitApplicable @return
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchBundleBankPolicy	<p>This method will fetch the bundle bank policy definition for the input bank code. </p> @param bankCode @return	<p>This method makes a call to the BankPolicyDefinition and checks when the bundle benefits are to be applied to the account(s) which form part of the bundle. </p><p>This method can return ANY - indicating that bundle benefits are applicable if&nbsp;a account under any of the offer(s) belonging to the bundle are opened. </p><p>This method can return ALL - indicating that bundle benefits are applicable if account(s) under all mandatory offer(s) belonging to the bundle are opened. </p>
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	This will validate the maximum number of offset accounts that can be linked to a loan account. @param
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	Validates the roll-over policy for Loan Account Variation.
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDDABankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchDDABankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isEffectiveBundleExists	This method is used to get the status that whether there are effective bundles exist in the system currently or not.	This method is used to get the status that whether there are effective bundles exist in the system currently or not.
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountProductAdapter	isEffectiveBundleExists	This method is used to get the status that whether there are effective bundles exist in the system currently or not.	This is helper method to validate the bundle parameters in bank policy whether they can be updatable or not based  on the whether effective bundles exist or not in the system.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	isEffectiveBundleExists	This method is used to get the status that whether there are effective bundles exist in the system currently or not.	@param bundleBankPolicy
IAccountProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountProductAdapter	fetchHardshipPolicy	@param hardshipPolicyCode @param productClass @return	This method is used to fetched allowed relief types for populating the matrix. @param sessionContext @param accountId @param hardshipReliefTypes @return @throws FatalException
ILimitsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAdapter	getFacilityCategoryFromProductGroupCode	This method is used to get the facility Category for the given Product Group Code and Business Unit Code. @param productGroupCode @param businessUnitCode @return facilityCategoryCode @throws FatalException	This method is used to get the facility category code for the given business unit and product group code. @param sessionContext @param productGroupCode @param businessUnitCode @return facilityCategoryCode
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	fetchDocumentDetails	This method will fetch the document details from document module depending upon the document type id. @param sessionContext @param documentTypeId @return DocumentTypeDTO	
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	fetchDocumentDetails	This method will fetch the document details from document module depending upon the document type id. @param sessionContext @param documentTypeId @return DocumentTypeDTO	This method will fetch the document details from document module depending upon the document type id. @param sessionContext @param documentTypeId @return DocumentTypeDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	fetchDocumentDetails	This method will fetch the document details from document module depending upon the document type id. @param sessionContext @param documentTypeId @return DocumentTypeDTO	This method will fetch document description for the specified docTypeId. @param sessionContext @param docTypeId @return String
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	getInboundDocumentChecklist	This service will fetch the list of inbound documents grouped under a checklist.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service will fetch the list of inbound documents grouped under a checklist.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	getInboundDocumentChecklist	<p>This service will fetch the list of inbound documents grouped under a checklist.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException</p>	<p>This method is used to update document list for new Associated Party @param sessionContext @param associatedPartyDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	getInboundDocumentChecklist	This service will fetch the list of inbound documents grouped under a checklist.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service creates the document checklist for associated party  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,AssociatedPartyDTO,associatedPartyDTO, ,associatedPartyDTO TODO: Explain the associatedPartyDTO above @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException @exception   {@link CardErrorConstants#INBOUND_DOC_CHKLIST_FAILED} : OBP_CC_0007   This exception is thrown if there is a problem in inbound document checklist generation.    
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	updateDocumentChecklist	This service updates the document checklist with associated party id. @param sessionContext @param documentChecklistDTO @throws FatalException	This service updates the document checklist with associated party id. @param sessionContext @param documentChecklistDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	updateDocumentChecklist	This service updates the document checklist with associated party id. @param sessionContext @param documentChecklistDTO @throws FatalException	This represents the addAssociatedParty method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,AssociatedPartyDTO,associatedPartyDTO, ,associatedPartyDTO TODO: Explain the associatedPartyDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDocumentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDocumentAdapter	updateDocumentChecklist	This service updates the document checklist with associated party id. @param sessionContext @param documentChecklistDTO @throws FatalException	This method is used to update document list for new Associated Party @param sessionContext @param associatedPartyDTO @throws FatalException
IDDACardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACardAdapter	fetchCardNumbersLinkedToAccount	This method is defined to get the list of credit card numbers for which the CASA acts as a payment account. @param accountId @return	
ITDELCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDELCMAdapter	updateELCMLienDetails	@param sessionContext @param tdELCMLienDetailsDTO	
ITDELCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDELCMAdapter	updateELCMLienDetails	@param sessionContext @param tdELCMLienDetailsDTO	This method calls the ELCM module to send the lien details to the ELCM module. @param accountId @param depositId @param renewDepositId @param renewalAmount @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>of accounting event &lt;br&gt; into event table  &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt;  &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt;  &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls  &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofs.s.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus.  &lt;/pre&gt; @param accountingEventDTO @throws FatalException</p>	<pre> traits()&lt;br /&gt; &lt;br /&gt; &lt;li&gt;Validate the loan account via validateLoanAccount() method.&lt;/li&gt; &lt;li&gt;Fetch the customer loan account via LoanAccountService.fetchLoanAccount() method&lt;/li&gt; &lt;li&gt;Fetch the loan product preferences associated with the customer loan account via LoanProductApplication.fetchLoanProduct() method.&lt;/li&gt; &lt;li&gt;Increase the unclear amount on successful cheque posting/outgoing collection request for external account via LoanAccount.increaseUnclearAmount() method&lt;br /&gt;&lt;/li&gt; &lt;li&gt;Raise Accounting only when preference at product level is bookBalance, incase of collectedBalance accounting will be raised by payments module &lt;br /&gt; &lt;/li&gt; &lt;/ol&gt; &lt;blockquote&gt; if(!LoanProductPreferencesDTO.collectedBalance) &lt;br /&gt; { &lt;br /&gt; //Prepare an AccountingEventDTO for crediting unclear amount(Cr LN_UNCLEARFUNDS) and Debit Clearing Bridge GL(Dr CLEARING_BRIDGE_GL) via LoanAccountingService.prepareAccountingEventDTO() method&lt;br /&gt; &lt;br /&gt; //Raise accounting event passing the AccountingEventDTO via LoanAccountingService.raiseAccountingEvent()&lt;br /&gt; &amp;nbsp;}&lt;/blockquote&gt; @param sessionContext @param accountId @param transactionDetailsDTO @param loanEventCode @return @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	<pre> @param accountId @param isChequeTransaction @param applicationContext @param loanProductPreferenceDTO @param amountInAccountCurrency </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &lt;br&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &lt;br&gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	<p>This method constructs an {@link AccountingEventDTO} and raises an accounting event. NOTE:This method is specifically created for the method manageFundsOnStandingInstructionExecution as one of the account role's (LN_SI_BRIDGE) for raising the accounting event is module supplied and ledger code need to be derived for the same. @param sessionContext @param accountId @param applicationContext @param loanAccount @param amountInAccountCurrency @param loanAccountingService @return amountTagDetailsDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> &lt;pre&gt;This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred.&lt;/br&gt; After getting the required details for processing of accounting entries from different modules &lt;/br&gt; of application, this method will persist the input values of accounting event &lt;br&gt; into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; &gt; which will have one to one relationship with behaviour &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;br&gt; &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;.&lt;/br&gt;This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} &lt;br&gt; which will return the TransactionStatus. &lt;/pre&gt; @param accountingEventDTO @throws FatalException </pre>	<pre> This method is responsible for Raising a Single Accounting Event. It internally calls {@link LoanAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.accounting.dto.AccountingEventDTO)} &lt;br&gt; which returns the TransactionStatus. &lt;/pre&gt; @param AccountingEventDTO &lt;pre&gt; This DTO contains the information of the Accounting event that has to be raised &lt;/pre&gt; @param applicationContext @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseAccountingEvent	<pre> This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries from different modules of application, this method will persist the input values of accounting event into event table &lt;b&gt;FLX_DA_ACCT_EVENT_B&lt;/b&gt; which will have one to one relationship with behaviour &lt;b&gt;FLX_DA_ACCT_BEHAVIOUR_B&lt;/b&gt; and one to many relationship with details &lt;b&gt;FLX_DA_ACCT_EVENT_DTLS_B&lt;/b&gt;. This method calls {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus. @param accountingEventDTO @throws FatalException </pre>	<p>This method is responsible for Raising Multiple Accounting Events. It internally calls {@link LoanAccountingAdapter#raiseAccountingEvent(com.ofss.fc.app.accounting.dto.AccountingEventDTO)} in an iteration which returns the TransactionStatus.</p> <p>This method can be helpful in following scenarios:-  Example: Service 'A' invokes Service 'B'.  Caller: Service A  Being called: Service B  Service 'A' has essential details to raise an Accounting Event, where as Service 'B' do not have the same. In such cases, service B would just prepare the Accounting Event with available details and would send it back to caller.  Caller would prepare the entire set of Accounting Event DTOs and would raise them together.</p> <p>@param Set &lt;AccountingEventDTO&gt; This contains a set of AccountingEventDTO  @param applicationContext  @throws FatalException</p>
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseReverseAccountingEvent	<pre> This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal </pre>	<pre> This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. This method prepares {@link AccountingBehaviourDTO} and following are set. accountingBehaviourDTO.setIsAuthorized(true); </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>event is created. &lt;/br&gt; This method prepares {@link AccountingBehaviourDTO} and following are set.</p> <pre>accountingBehaviourDTO.setIsAuthorized(true); accountingBehaviourDTO.setIsBalancingRequired(true);</pre> <p>&lt;/br&gt; Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus.</p> <pre>&lt;/pre&gt; @param sequenceNumberSequence number of the current accounting entry @param transactionReferenceNumberCurrent transaction reference number @param originalTransactionReferenceNumberOriginal transaction reference number @param loanEventConstantLoan Event Constant @throws FatalException</pre>	<pre>accountingBehaviourDTO.setIsBalancingRequired(true);</pre> <p>&lt;/br&gt; Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseAndProcessReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus.</p> <pre>&lt;/pre&gt; @param sequenceNumberSequence number of the current accounting entry @param transactionReferenceNumberCurrent transaction reference number @param originalTransactionReferenceNumberOriginal transaction reference number @param loanEventConstantLoan Event Constant @throws FatalException</pre>
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	raiseReversalAccountingEvent	<pre>&lt;pre&gt;This method is used for raising a reversal event for the transaction and event sequence no provided. &lt;/br&gt; Using the originalReferenceNo and originalEventSequenceNo, the</pre>	<pre>&lt;p&gt;This method is responsible for raising a Reversal Accounting Event for the entire transaction. Using the originalReferenceNo (Original Transaction Reference Number) and originalEventSequenceNo (Event Sequence Number of Original Transaction), &lt;br&gt; an Accounting Event for Reversal is created.&lt;br&gt;This</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>details of the event to be reversed is obtained &lt;br&gt; and a new reversal event is created. &lt;/br&gt; This method prepares {@link AccountingBehaviourDTO} and following are set.</p> <pre>accountingBehaviourDTO.setIsAuthorized(true); accountingBehaviourDTO.setIsBalancingRequired(true);</pre> <p>&lt;/br&gt; Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus.</p> <pre>&lt;/pre&gt; @param sequenceNumberSequence number of the current accounting entry @param transactionReferenceNumberCurrent transaction reference number @param originalTransactionReferenceNumberOriginal transaction reference number @param loanEventConstantLoan Event Constant @throws FatalException</pre>	<p>method internally invokes Accounting Module Services via {@link LoanAccountingAdapter#raiseReverseAccountingEvent(Integer sequenceNumber, String transactionReferenceNo, String originalTransactionReferenceNo,String loanEventConstant)} &lt;/p&gt; @param sequenceNumber @param transactionReferenceNo @param originalTransactionReferenceNo @param LoanEventConstant @param applicationContext @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	reverseAccounting	@param loanEvent @param refNum	@param eventwiseTransactionsForReversal
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	fetchMaxEventSequenceNumber		
ILoanAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAccountingAdapter	fetchMaxEventSequenceNumber		
IRoleEvaluationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IRoleEvaluationAdapter	isAdapterMocked	This methods checks whether adapter mocking is enabled ; If true, it returns true to adapter mocked. If false, it returns the adapter mocking flag for the adapter from the adapter factory properties. @param adapterMockConstant @return boolean	This method evaluates the role that has to be used. @throws FatalException
IRoleEvaluationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IRoleEvaluationAdapter	fetchDefaultRole	This method reads a property file and fetches the Property value  for given property name. @fcb.param MI,String, propertyName, contains the name of the property  for which value needs to be fetched.  @fcb.param MI,String, defaultPropertyName, , contains the default property name.  @return String @throws FatalException	This method reads a property file and fetches the Property value  for given property name. @fcb.param MI,String, propertyName, contains the name of the property  for which value needs to be fetched.  @fcb.param MI,String, defaultPropertyName, , contains the default property name.  @return String @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IRoleEvaluationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IRoleEvaluationAdapter	fetchDefaultRole	This method reads a property file and fetches the Property value  for given property name. @fcb.param MI,String, propertyName, contains the name of the property  for which value needs to be fetched. @fcb.param MI,String, defaultPropertyName, , contains the default property name. @return String @throws FatalException	This method evaluates the role that has to be used. @throws FatalException
IRoleEvaluationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IRoleEvaluationAdapter	fetchDefaultRole	This method reads a property file and fetches the Property value  for given property name. @fcb.param MI,String, propertyName, contains the name of the property  for which value needs to be fetched. @fcb.param MI,String, defaultPropertyName, , contains the default property name. @return String @throws FatalException	
IRoleEvaluationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IRoleEvaluationAdapter	fetchDefaultRole	This method reads a property file and fetches the Property value  for given property name. @fcb.param MI,String, propertyName, contains the name of the property  for which value needs to be fetched. @fcb.param MI,String, defaultPropertyName, , contains the default property name. @return String @throws FatalException	This method retrieves primary application role from the given role list.  If given role list contains anonymous-role, it removes this role from list and  then searches for primary application roles.  If it finds multiple primary roles then it returns first primary role in the list.  If it doesn't find any primary role or it encounter any error then it returns first  role from given role list.  If given role list only contains anonymous-role role then it returns default role.  @param roleList @return String @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IDispatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IDispatchAdapter	dispatchSMS		
IDispatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IDispatchAdapter	dispatchSMS		
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the endpoint leg for accounting event. @param transactionContext @param transactionLog @param sequenceNumber @param accountingEventAmountTagDetailsDTO @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	{@link OutgoingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense @see com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				entity .log.TransactionLog)
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the accounting leg and triggers accounting for Business Collections. </br> @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param messageLogItem @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the endpoint leg for accounting event. @param transactionLog @param sequenceNumber @param accountingEventAmountTagDetailsDTO @param endpointLedgerId @param drAmountTag @param instrumentLogItem @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	{@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense {@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)} @param transactionContext @param transactionLog

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the endpoint leg for accounting event. @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param messageLogItem TODO @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	for more info {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. <style type="text/css">table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;&amp;nbsp;&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PRECUTOFF&lt;/td&gt; &lt;td&gt;C54&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;E58&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PSTCUTOFF&lt;/td&gt; &lt;td&gt;C55&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E59&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PRECUTOFF&lt;/td&gt; &lt;td&gt;E60&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;E61&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PSTCUTOFF&lt;/td&gt; &lt;td&gt;E62&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E63&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_MOVEMENT_HELDOVR_ENDPOINT&lt;/td&gt; &lt;td&gt;E64&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E65&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_PRECUT&lt;/td&gt; &lt;td&gt;E66&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;C56&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_POSTCUT&lt;/td&gt; &lt;td&gt;E67&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C57&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from GL&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_GL&lt;/td&gt; &lt;td&gt;E68&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E69&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_RECALL_MVMT_HELDOVR_ENDPT&lt;/td&gt; &lt;td&gt;E70&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E71&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment for ASP Verification&lt;/td&gt; &lt;td&gt;PY_OUTPAY_ASP_VERIFY&lt;/td&gt; &lt;td&gt;E75&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_MISC_EXP&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E76&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param transactionLog @param amountTagDetailsDTO @param messageLogItem @throws FatalException </pre>
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	For accounting {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	for more info { <a href="#">@link</a> OutgoingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)} @param transactionContext @param transactionLog @param amountTagDetailsDTO @param eventId @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the accounting leg and triggers accounting for Business Payments. </br> @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param messageLogItem @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	It it undo the transaction and raise dishonor accounting. @param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method will do the re-process accounting of refused transactions. @param transactionContext @param transactionLog @param messageLogItem @param reverse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method will do the refusal accounting of the dishonor of outward transactions @param transactionContext @param transactionLog @param messageLogItem @param reverse TODO @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method provides the common suspense handling mechanism. A suspense can happen, when the transaction must honored inspie of failure. This will happen if the transaction is posted as advice (force post) @param transactionContext TODO @param messageItem TODO @param txnLog @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	@param transactionLog @param messageLogItem @param isCredit @param sequenceNumber @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method makes a call to the payments product adapter to fetch the endpoint ledger for raising the accounting event. @param transactionContext @param transactionLog @param messageLogItem TODO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	This method builds the endpoint leg for accounting event. @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTOList @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchEndpointLedger	This method will fetch the endpoint ledger for the supplied endpoint id. @param fundExchangeType @param collectionType @param endpointId @return @throws FatalException	@see IncomingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem) com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.entity .log.TransactionLog)
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchHeldoverLedger	This method will fetch the heldover ledger. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchHeldoverLedger	This method will fetch the heldover ledger. @return @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. <style type="text/css"> table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} </style> <table class="tableizer-table"> <tr class="tableizer-firstrow"> <th>Actual Event</th> <th>Event Code</th> <th>Dr Transaction Code</th> <th>Netting Indicator</th> <th>Account Role Code - Debit</th> <th>Amount Tag - Debit</th> <th>Cr Transaction Code</th> <th>Account Role Code - Credit</th> <th>&nbsp;</th> <th>Module responsibility</th> <th>GAAP Code</th> </tr>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PRECUTOFF&lt;/td&gt; &lt;td&gt;C54&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E58&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PSTCUTOFF&lt;/td&gt; &lt;td&gt;C55&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E59&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PRECUTOFF&lt;/td&gt; &lt;td&gt;E60&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E61&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PSTCUTOFF&lt;/td&gt; &lt;td&gt;E62&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E63&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_MOVEMENT_HELDOVR_ENDPOINT&lt;/td&gt; &lt;td&gt;E64&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E65&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_PRECUT&lt;/td&gt; &lt;td&gt;E66&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;C56&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_POSTCUT&lt;/td&gt; &lt;td&gt;E67&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C57&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from GL&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_GL&lt;/td&gt; &lt;td&gt;E68&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E69&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Heldover to endpoint movement&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;PY_RECALL_MVMT_HELDOVR_ENDPT&lt;/td&gt; &lt;td&gt;E70&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E71&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment for ASP Verification&lt;/td&gt; &lt;td&gt;PY_OUTPAY_ASP_VERIFY&lt;/td&gt; &lt;td&gt;E75&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_MISC_EXP&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E76&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param transactionLog @param amountTagDetailsDTO @param messageLogItem @throws FatalException </pre>
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchHeldoverLedger	This method will fetch the heldover ledger. @return @throws FatalException	For accounting {@link IncomingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchHeldoverLedger	This method will fetch the heldover ledger. @return @throws FatalException	This method will do the re-process accounting of refused transactions. @param transactionContext @param transactionLog @param messageLogItem @param reverse @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchHeldoverLedger	This method will fetch the heldover ledger. @return @throws FatalException	This method makes a call to the payments product adapter to fetch the heldover ledger for raising the heldover accounting event. @param transactionContext @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchClearingBridgeLedger	This method will fetch the clearing bridge ledger. @return @throws FatalException	This will return the Bridge GI for respective mode of payment. @param context @param accountTransactionDTO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchClearingBridgeLedger	This method will fetch the clearing bridge ledger. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchClearingBridgeLedger	This method will fetch the clearing bridge ledger. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	This method builds fetches the ledger code to be debited of credit during the processing of Business Transfer.   @fcb.param MI, NonProductDomainCategory, nonProductDomainCategory <li> {@link com.ofss.fc.enumeration.productmanufacturing.NonProductDomainCategory}</li> @fcb.param MI, String, accountRole @return ledgerId @throws BusinessException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	@param transactionContext @param transactionLog @param amountTagDetailsDTO @param messageLogItem @param sequenceNumber @param finalTagDTO @param accountingEventAmountTagDetailsDTO @param eventId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	This method is used to pay any prior scheduled Utility Bill.It will take scheduled paymentId,Bill amount and Bill date as input parameter and will generate Bill payment response. @param eventId @param context @param partyID @param scheduledPaymentID @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	This will return the Bridge GI for respective mode of payment. @param context @param accountTransactionDTO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	This method sends an outgoing payment of 0.01 cents to the counter party bank account for verification of its existence and sends the verification code(as narrative) as part of the outgoing payment. @param transactionContext @param verificationCode @param accountId @param financialInstitutionIdentificationDTO @return FundsExchangeResponse
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	<p>This method will build the heldover debit and credit legs for post-cutoff outgoing clearing transactions. The heldover leg is dervedid, so only some basic information needs to be added. </p> @param transactionLog @param isHeldoverCredit @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchMISCPaymentsLedger	returns the ledgerCode for the given domain category and account role @param domainCategory @param accountRole @return @throws FatalException	This method builds fetches the ledger code to be debited of credit during the processing of Business Transfer.   @fcb.param MI, NonProductDomainCategory, nonProductDomainCategory <li> {@link com.ofss.fc.enumeration.productmanufacturing.NonProductDomainCategory}</li> @fcb.param MI, String, accountRole @return ledgerId @throws BusinessException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This service validate the policy of immediate credit as well as perform the Immediate credit on provided data. It will raise an accounting event for the credit and also audit the transaction @throws FatalException @see com.ofss.fc.app.pc.core.IOutgoingCollectionService#doImmediateCredit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.dto.log.InstrumentTransferTransactionLogDTO, com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO, com.ofss.fc.datatype.Money)
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will do the funds realization associated with the instrument based on the event, and nature of transaction. @fcb.param transactionLog @fcb.param transactionEventType @fcb.param messageLogItem



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																																	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	<p>This method will raise the accounting event for successful outgoing collection.&lt;br&gt; The event is {<a href="#">@link PCAccountingConstants#EVT_IN_COLL</a>}. In this event, there is no accounting on the account's ledger. Only the CFC and endpoint Ledger are affected.</p> <pre> &lt;table border="1" style="border-collapse: collapse; width: 100%; font-family: Arial, Helvetica, sans-serif; font-size: 12px;"&gt; <thead> <tr> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Actual Event</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Event Code</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Dr Transaction Code</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Netting Indicator</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Account Role Code - Debit</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Amount Tag - Debit</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Cr Transaction Code</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Account Role Code - Credit</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Amount Tag - Credit</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">Module responsibility</th> <th style="padding: 4px; margin: 3px; border: 1px solid #ccc; background-color: #104E8B; color: #FFF; font-weight: bold;">GAAP Code</th> </tr> </thead> <tbody> <tr> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">Outward Collection of CASA for posting</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">CT_OUT_COLL_CASA_POST</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">E29</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">N</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">COLLECTIONS_ENDPOINT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">OUTWARD_COLLECTION_AMOUNT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">E30</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">COLLECTIONS_CFC</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">OUTWARD_COLLECTION_AMOUNT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">Collections</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">AL</td> </tr> <tr> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">Outward Collection for CASA realisation</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">CT_OUT_COLL_CASA_REALIZE</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">E31</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">N</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">COLLECTIONS_CFC</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">OUTWARD_COLLECTION_AMOUNT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">C48</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">COLLECTIONS_ACCOUNT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;">OUTWARD_COLLECTION_AMOUNT</td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;"></td> <td style="padding: 4px; margin: 3px; border: 1px solid #ccc;"></td> </tr> </tbody> </pre>	Actual Event	Event Code	Dr Transaction Code	Netting Indicator	Account Role Code - Debit	Amount Tag - Debit	Cr Transaction Code	Account Role Code - Credit	Amount Tag - Credit	Module responsibility	GAAP Code	Outward Collection of CASA for posting	CT_OUT_COLL_CASA_POST	E29	N	COLLECTIONS_ENDPOINT	OUTWARD_COLLECTION_AMOUNT	E30	COLLECTIONS_CFC	OUTWARD_COLLECTION_AMOUNT	Collections	AL	Outward Collection for CASA realisation	CT_OUT_COLL_CASA_REALIZE	E31	N	COLLECTIONS_CFC	OUTWARD_COLLECTION_AMOUNT	C48	COLLECTIONS_ACCOUNT	OUTWARD_COLLECTION_AMOUNT		
Actual Event	Event Code	Dr Transaction Code	Netting Indicator	Account Role Code - Debit	Amount Tag - Debit	Cr Transaction Code	Account Role Code - Credit	Amount Tag - Credit	Module responsibility	GAAP Code																											
Outward Collection of CASA for posting	CT_OUT_COLL_CASA_POST	E29	N	COLLECTIONS_ENDPOINT	OUTWARD_COLLECTION_AMOUNT	E30	COLLECTIONS_CFC	OUTWARD_COLLECTION_AMOUNT	Collections	AL																											
Outward Collection for CASA realisation	CT_OUT_COLL_CASA_REALIZE	E31	N	COLLECTIONS_CFC	OUTWARD_COLLECTION_AMOUNT	C48	COLLECTIONS_ACCOUNT	OUTWARD_COLLECTION_AMOUNT																													

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Collection for CASA return pre- realisation&lt;/td&gt; &lt;td&gt;CT_OUTCOLLCASA_RET_PRE_REALIZE&lt;/td&gt; &lt;td&gt;E32&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_CFC&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;E33&lt;/td&gt; &lt;td&gt;COLLECTIONS_ENDPOINT&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Collection for CASA return post- realisation&lt;/td&gt; &lt;td&gt;CT_OUTCOLLCASA_RET_PST_REALIZE&lt;/td&gt; &lt;td&gt;C49&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT_EQVLN T&lt;/td&gt; &lt;td&gt;E34&lt;/td&gt; &lt;td&gt;COLLECTIONS_ENDPOINT&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Collection for TD realisation&lt;/td&gt; &lt;td&gt;CT_OUT_COLL_TD_REALIZE&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;COLLECTIONS_CFC&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;CLEARING_BRIDGE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_COLLECTION_AMOUNT&lt;/td&gt; &lt;td&gt;Collections&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @see AbstractFundsExchangeServiceProvider#triggerAccounting(TransactionLog, AccountingEventAmountTagDetailsDTO[]) </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. <style type="text/css"> table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} </style> <table class="tableizer-table"> <tr class="tableizer-firstrow"> <th>Actual Event</th> <th>Event Code</th> <th>Dr Transaction Code</th> <th>Netting Indicator</th> <th>Account Role Code - Debit</th> <th>Amount Tag - Debit</th> <th>Cr Transaction Code</th> <th>Account Role Code - Credit</th> <th>Amount Tag - Credit</th> <th>Module responsibility</th> <th>GAAP Code</th> </tr> <tr> <td>Cheque Deposit</td> <td>CL_CHQ_DEPOSIT_TO_CASA</td> <td>E01</td> <td>N</td> <td>CLEARING_ENDPOINT</td> <td>CHEQUE_AMOUNT_EQUIVALENT</td> <td>E02</td> <td>CLEARING_CFC</td> <td>CHEQUE_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> <tr> <td>Immediate Credit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> (only to CASA accounts) &lt;td&gt;CL_CHQ_IMM_CREDIT_TO_CASA &lt;td&gt;E03&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT &lt;td&gt;C63&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation with partial immediate credit &lt;td&gt;CL_CHQ_REALIZE_PARTIAL_IMM_CR &lt;td&gt;E04&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC &lt;td&gt;BALANCE_CHEQUE_AMOUNT &lt;td&gt;C40&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT &lt;td&gt;BALANCE_CHEQUE_AMOUNT &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation without immediate credit &lt;td&gt;CL_CHQ_REALIZE_NO_IMM_CREDIT &lt;td&gt;E05&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C41&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to TD account &lt;td&gt;CL_CHQ_REALIZE_TO_TD &lt;td&gt;E06&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;T64&lt;/td&gt; &lt;td&gt;CLEARING_TDACCOUNT &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to Loans account&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_TO_LOAN&lt;/td&gt; &lt;td&gt;E07&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;L01&lt;/td&gt; &lt;td&gt;CLEARING_LOANACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to Biller&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_TO_BILLER&lt;/td&gt; &lt;td&gt;E08&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;F01&lt;/td&gt; &lt;td&gt;RESPECTIVE_BILLER_GL&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Cheque Realisation to GL&lt;/td&gt; &lt;td&gt;CL_CHQ_REALIZE_TO_GL&lt;/td&gt; &lt;td&gt;E09&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E78&lt;/td&gt; &lt;td&gt;CLEARING_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Dishonour of cheque before value date&lt;/td&gt; &lt;td&gt;CL_CHQ_DISHONOR_PRE_VALUEDATE&lt;/td&gt; &lt;td&gt;E10&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E11&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Dishonour of cheque after value date&lt;/td&gt; &lt;td&gt;CL_CHQ_DISHONOR_POST_VALUEDATE&lt;/td&gt; &lt;td&gt;C42&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E12&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Reject of O/w Clearing&lt;/td&gt; &lt;td&gt;CL_OUTWARD_CHQ_REJECT&lt;/td&gt; &lt;td&gt;E13&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E14&lt;/td&gt; &lt;td&gt;OUTWARD_REJECT_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Clearing Reject to Pass&lt;/td&gt; &lt;td&gt;CL_OUTCHQ_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;E15&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;OUTWARD_REJECT_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E16&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Return of partial immediate credit&lt;/td&gt; &lt;td&gt;CL_RETURN_CHQ_PART_IMM_CREDIT&lt;/td&gt; &lt;td&gt;C43&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT&lt;/td&gt; &lt;td&gt;E17&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;IMMEDIATE_CREDIT_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Return of partial immediate credit&lt;/td&gt; &lt;td&gt;CL_RETURN_CHQ_PART_IMM_CREDIT&lt;/td&gt; &lt;td&gt;E18&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CFC&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E19&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<td>CHEQUE_AMOUNT</td> <td>Clearing</td> <td>AL</td> </tr> </table> @param transactionLog @param accountingEventAmountTagDetailsDTOArray @throws FatalException @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will do the validation and any pre-processing that is required for the transaction. All business policy violations arising put of validations will be updated with the transaciton and the transaction itself will be sent for authorization. @param transactionCategory @param transactionLog @param context @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	@see com.ofss.fc.app.pc.core.IOutgoingClearingService#validate(com.ofss.fc.app.context.SessionContext, java.lang.String, com.ofss.fc.app.pc.dto.input.InstrumentTransferCaptureDTO)
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	It it undo the transaction and raise dishonor accounting. @param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will be called when the original transaction is found. It will make a call to the account module to undo the account transaction. If the account transaction is successfully undone, raise dishonour accounting and log details. If the account transaction fails, raise refusal accounting and log details. @param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	<p>&lt;p&gt;This method will validate the account before trying to debit the account. It will delegate the call to the appropriate module to which the account belongs &lt;/p&gt;</p> <p>&lt;p&gt; First , the system will check if the account type is available or not in the transaction log. &lt;br&gt; If not then, the system will determine which module does the given account belongs to by &lt;/p&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;making a call to @link com.ofss.fc.app.account.service.core.AccountInquiryApplicationService.</li> <li>&lt;li&gt;fetchAccountByAccountType(SessionContext, String, AccountType)&lt;/li&gt;</li> <li>&lt;li&gt;From the fetched dto get the account type.&lt;/li&gt;</li> <li>&lt;li&gt;Converting the Account type to TaskModuleType&lt;/li&gt;</li> <li>&lt;li&gt;Update the log with the TaskModuleType&lt;/li&gt;</li> </ul> <p>&lt;p&gt; and then call the appropriate module adapter based on the TaskModuleType &lt;/p&gt;</p> <p>@param transactionContext @param accountId @param transactionLog @param messageLogItem @return @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will check if for a given product, CFC accounting is required or not. @param transactionLog @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	<p>&lt;p&gt;This method will validate the account before trying to credit the account. It will delegate the call to the appropriate module to which the account belongs &lt;/p&gt; &lt;p&gt; First , the system will check if the account type is available or not in the transaction log. &lt;br&gt; If not then, the system will determine which module does the given account belongs to by &lt;/p&gt; &lt;ul&gt; &lt;li&gt;making a call to @link com.ofss.fc.app.account.service.core.AccountInquiryApplicationService.fetchAccountByAccountType(SessionContext, String, AccountType)&lt;/li&gt; &lt;li&gt;From the fetched dto get the account type.&lt;/li&gt; &lt;li&gt;Converting the Account type to TaskModuleType&lt;/li&gt; &lt;li&gt;Update the log with the TaskModuleType&lt;/li&gt; &lt;/ul&gt; &lt;p&gt; and then call the appropriate module adapter based on the TaskModuleType &lt;/p&gt; @param transactionContext @param accountId @param transactionLog @param messageLogItem @return ValidationResultDTO @throws FatalException</p>
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	This method will allow the system to reverse the transaction that has already been submitted and dispatched. Reversal is usually done by bank staff, when the mistake is done by staff. No penalty interest is charged here. @param transactionContext @param fundExchangeType @param collectionType @param transactionReferenceNo @param subSequenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccounting	This method checks if the product indicator has BookBalance or CollectedBalance. @return @throws FatalException	@param transactionContext @param originalTransactionLog @param transactionLoggerService @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccountingForLoanAccount	This method checks if the product indicator has BookBalance or CollectedBalance for Loan Account. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	checkForCFCAccountingForLoanAccount	This method checks if the product indicator has BookBalance or CollectedBalance for Loan Account. @return @throws FatalException	This method will check if for a given product, CFC accounting is required or not. @param transactionLog @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	This method is used to override the accounting tag details which is received after settlement with MC accounting tags. the tag details that are received by settlement has only debit lag , here we will create a MC credit lag also and hence the list of tag details will now have 2 elements. one casa debit leg and MC GL credit leg. @param IsSettlementSource @param entity entity has all the information related to bankDraft (Settlement mode, Amount, serial-number, etc) @param listTagDTO . Accounting tag details.
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	This method will creates the settlement bridgeGL accounting leg when bank draft is issue via settlement. @param postingDate @param bankDraft @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchSettlementBridgeGL	This method will fetch the settlement bridge GL ledger. @return @throws FatalException	This method will derive the accounting leg for the bridgeGL.This is also calculated from the ban draft liquidation perspective @param transactionContext @param bankDraft @param creditAmountTag @param sequenceNumber @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchIntermediaryBundleGL	A new service will be provided by the PMU to fetch the intermediary bundle GL and the collections module will have to call for this service to fetch the GL. @return @throws FatalException	This method will populate the credit leg when the caller is BUNDLE. @fcb.param transactionLog @fcb.param messageLogItem @fcb.param amountTagId @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchIntermediaryBundleGL	A new service will be provided by the PMU to fetch the intermediary bundle GL and the collections module will have to call for this service to fetch the GL. @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	fetchIntermediaryBundleGL	A new service will be provided by the PMU to fetch the intermediary bundle GL and the collections module will have to call for this service to fetch the GL. @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting event dto for raising the accounting event.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, String, eventId @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @return {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO} @throws BusinessException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. @fcb.param transactionLog @fcb.param accountingEventAmountTagDetailsDTOArray

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting dto for transactions. @param transactionContext @param transactionLog @param finalTagDTO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	{@link OutgoingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense @see com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.entity .log.TransactionLog)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	<pre> &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;Amount Tag - Credit&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Debit&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_DEBIT&lt;/td&gt; &lt;td&gt;C44&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT_EQUIVALENT&lt;/td&gt; &lt;td&gt;E20&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT&lt;/td&gt; &lt;td&gt;E21&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E22&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Pass to Reject&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_PASS_TO_REJECT&lt;/td&gt; &lt;td&gt;E23&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;C45&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;I/w Clearing Reject to Pass&lt;/td&gt; &lt;td&gt;CL_INWARD_CHQ_REJECT_TO_PASS&lt;/td&gt; &lt;td&gt;C46&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_CASAACCOUNT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E24&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;O/w Clearing for I/w Clearing Reject&lt;/td&gt; &lt;td&gt;CL_OUTWARD_INWARD_REJECTS&lt;/td&gt; &lt;td&gt;E25&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E26&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Float Extension&lt;/td&gt; &lt;td&gt;CL_CHQ_FLOAT_EXTENSION&lt;/td&gt; &lt;td&gt;E27&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;CLEARING_ENDPOINT&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;E28&lt;/td&gt; &lt;td&gt;INWARD_CLEARING_SUSPENSE&lt;/td&gt; &lt;td&gt;CHEQUE_AMOUNT&lt;/td&gt; &lt;td&gt;Clearing&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; </pre>
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting dto for transactions. @param transactionContext @param transactionLog @param accountingEventAmountTagDetailsDTO @param eventId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	{@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method raise accounting while standing instruction batch execution and the biller is consolidated. @param applicationContext @param accountingEventAmountTagDetailsDTOs @param transactionAmount @param narration @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting event dto. @param applicationContext @param eventBillPay @param remitComponentPreferenceItemDTO @param listAccountingEventAmountTagDetailsDTO @param transactionAmount @param transactionDescription @param isNotConsolidated @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will populate the accounting event dto for raising accounting for tc encashment. @param applicationContext TODO @param eventId @param remarks @param AccountingEventAmountTagDetailsDTO [] <li>{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO} @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting dto for pre cut-off transactions. @param transactionContext @param transactionLog @param relatedAccountStatus @param finalTagDTO @param messageLogItem TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	for more info <a href="#">{@link OutgoingPaymentService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}</a> @param transactionContext @param transactionLog @param amountTagDetailsDTO @param eventId @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	@param transactionContext @param transactionLog @param messageLogItem TODO @param eventId @param finalTagDTO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	Method to do debit accounting for consolidated bill payment @param accountingEventAmountTagDetailsDTOs @param transactionAmount @param narration @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This service will call prepareAccountingLeg which will be used to override the accounting tag details which is received after settlement with MC accounting tags.  <li>It will also raise the accounting for the bank draft issuance. <li>If bank draft is issue via settlement then build the settlement bridge GL leg @param accountingLegs . Accounting tag details. @param IsSettlementSource @param entity entity has all the information related to bankDraft (Settlement mode, Amount, serialnumber, etc)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will raise an accounting for the bank draft liquidation only. @param transactionContext @param bankDraft @param eventId @param accountingEventAmountTagDetailsDTO @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will do the refusal accounting of the dishonor of outward transactions @param transactionContext @param transactionLog @param messageLogItem @param reverse TODO @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method will call PMU service to determine if the event being raised is product event or non-product event. @param eventId @return @throws FatalException
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting dto for transactions. @param transactionContext @param transactionLog @param isSuspenseHandlingRequired @param finalTagDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsProductAdapter	isProductAccountingDomainCategory	Given the event id, this service will return if the accounting domain category for the event is product or non-product. Based on this information, accounting event to be sent for raise accounting is to be populated. @param eventId @return @throws FatalException	This method populates the accounting event dto for return transactions. @param transactionContext @param transactionLog @param eventId @param ledgerId @param eventAmountTagDetailsDTOs @param messageLogItem TODO @return @throws FatalException
IProductLedgerRuleApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductLedgerRuleApplicationAdapter	getRuleDetails	This method returns rule details on the basis of the rule Id passed. It returns all the information about a rule. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>, <b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>ruleId</b></u><b>:</b>  A valid rule Id for which EarMark and limit has to be checked  @param applicationContext @param ruleId @return ruleDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductLedgerRuleApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductLedgerRuleApplicationAdapter	getRuleDetails	This method returns rule details on the basis of the rule Id passed. It returns all the information about a rule. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>,<b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After  successful authentication the user request is processed. @fcb.param MI,String,<b><u>ruleId</b></u><b>:</b>  A valid rule Id for which EarMark and limit has to be checked  @param applicationContext @param ruleId @return ruleDTO	
IProductLedgerRuleApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductLedgerRuleApplicationAdapter	getRulesForSDEFacts		This represents the fetchRulesForSDEFacts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,LedgerAdditionalAttributesDTO,ledgerAdditionalAttributesDTO ,ledgerAdditionalAttributesDTO TODO: Explain the ledgerAdditionalAttributesDTO above @return {@link com.ofss.fc.service.response.BaseResponse.RuleSearchResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Failure</i> section
IntegrationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationAccountAdapter	fetchOffsetAccount	This represents the fetchOffsetAccount method of the entity. @return {@link com.ofss.fc.app.account.dto.offsetaccount.OffsetAccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IntegrationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationAccountAdapter	fetchOffsetAccount	This represents the fetchOffsetAccount method of the entity. @return {@link com.ofss.fc.app.account.dto.offsetaccount.OffsetAccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchOffsetAccount method of the entity. @return {@link com.ofss.fc.app.account.dto.offsetaccount.OffsetAccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IntegrationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationAccountAdapter	fetchOffsetAccount	This represents the fetchOffsetAccount method of the entity. @return {@link com.ofss.fc.app.account.dto.offsetaccount.OffsetAccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IntegrationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationAccountAdapter	fetchAssociatedPartyDetails	This returns the associatedPartyDTO for submission @param sessionContext @param associatedPartyDTO @return @throws com.ofss.fc.infra.exception.FatalException	This method provides the Referee details for the provided submission id. @param sessionContext @param submissionId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationAccountAdapter	fetchAssociatedPartyDetails	This returns the associatedPartyDTO for submission @param sessionContext @param associatedPartyDTO @return @throws com.ofss.fc.infra.exception.FatalException	This returns the associatedPartyDTO for submission @param sessionContext @param associatedPartyDTO @return @throws com.ofss.fc.infra.exception.FatalException
IJournalLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalLoanAdapter	fetchLoanAccount	This methods fetches the account for the accountId passed. @param accountId String @return account LoanAccount	
IJournalLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalLoanAdapter	fetchLoanAccount	This methods fetches the account for the accountId passed. @param accountId String @return account LoanAccount	
IJournalLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalLoanAdapter	fetchLoanAccount	This methods fetches the account for the accountId passed. @param accountId String @return account LoanAccount	
IPopulateGrantedValuesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPopulateGrantedValuesAdapter	checkGrantedValues	@param PopulateGrantedValuesDTO @return PopulateGrantedValuesResponse @throws FatalException	@param PopulateGrantedValuesDTO @return PopulateGrantedValuesResponse @throws com.ofss.fc.infra.exception.FatalException
IChannelConfigurationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelConfigurationAdapter	initiateChannelConfiguration	This method configures the currencyCache,transactionadppterconfig.parserfactory @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	Initialize all singletons used in the product . Currently supports , channel , businessobject, referenceobject, attribute definition , udf singletons. Please add more in this method @param sessionContext

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelConfigurationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelConfigurationAdapter	initiateEventPoller		Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller , delayed transaction poller , FTP webservice poller. Please add more in this method Additionally we also have an extension for consulting pollers @param sessionContext
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processNewRequest		processes a new request @see com.ofss.fc.app.adapter.lcm.collaterals.valuation.ICollateralValuationAdapter#processNewRequest(com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationObject)
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processNewRequest		@param SessionContext sessionContext : holds the basic information about the user of the system. @param collateralValuationDTO @param collateralValuation @throws FatalException @throws RuntimeException
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processCancelRequest		processes cancellation request. @return valuerAdapter @see com.ofss.fc.app.adapter.lcm.collaterals.valuation.ICollateralValuationAdapter#processCancelRequest(com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationCancellationDTO)
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processCancelRequest		This method <u>Cancels</u> the valuation request for the given request message reference . @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationCancellationDTO cancellationDTO : Collateral valuation request cancellation details DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processQueryRequest		processes a query request. @see com.ofss.fc.app.adapter.lcm.collaterals.valuation.ICollateralValuationAdapter#processQueryRequest(com.ofss.fc.app.lcm.dto.collaterals.valuation.report.query.CollateralValuationQueryDTO)
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processQueryRequest		This method will <u>initiate</u> a Query Message on a particular Collateral Valuation Report. The main details required for raising a Query are as follows.  Valuation Report Reference Query Comments. @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationQueryDTO collateralValuationQueryDTO : holds details of the collateral valuation query. @return void
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processRequestAmend		processes an amend request. @see com.ofss.fc.app.adapter.lcm.collaterals.valuation.ICollateralValuationAdapter#processRequestAmend(com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationObject)
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	processRequestAmend		
ICollateralValuationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationAdapter	validateReportDate	Date validation method added to validate if the report date is prior to the request date. This validation was earlier a part of product code. However, this has been included as part of an adapter interface so that it can be overridden.	This method is used to add response from third party interface or if mode is physical from collateral valuation report screen. @param aIValResponse @return aIValResponse @throws ParseException @throws FatalException @throws ClassNotFoundException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	getEventforEventId	This method fetches an instance of Event from the event cache @param domainId the domain id @param domainCategoryId the domain category id @param eventId Id of event to be loaded @return event {@link EventDTO} containing information about event	This method will return event name from event code and domain information to be used for retrieval from cache. @param domainId @param domainCategoryId @param eventId @return
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	getEventforEventId	This method fetches an instance of Event from the event cache @param domainId the domain id @param domainCategoryId the domain category id @param eventId Id of event to be loaded @return event {@link EventDTO} containing information about event	Helper method to fetch the event defined for the given domain, domain category and eventCode @return
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	publishPPCTemplateForProduct	This method will provide the product module with the array of {@link PPCTemplateDetailsDTO} which contains the information of events and the corresponding price used for the creation of PPC. @param sessionContext @param productGroup @param ppcTemplateDetailsDTOs @throws FatalException	This method will publish PPC configuration for fees. @param sessionContext @param pricePolicyChart @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllEventIds	Returns a list containing ids of all the events present in the repository, with the given domain and domain category. Ensure that inputs are non-null. @param domainId the domain id @param domainCategoryId the domain category id @return the Set of event ids	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllEventIds	Returns a list containing ids of all the events present in the repository, with the given domain and domain category. Ensure that inputs are non-null. @param domainId the domain id @param domainCategoryId the domain category id @return the Set of event ids	Given the pricing context, the policy chart level and the date, this method evaluates the appropriate margin index policy chart and returns it. Ensure that all inputs are non-null. No null-checks are performed. @param pricingContext the pricing context @param policyChartLevel the policy chart level @param evaluationDate the date for which to evaluate @return evaluated policy chart of type {@link RateOfferingElement}
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllEventIds	Returns a list containing ids of all the events present in the repository, with the given domain and domain category. Ensure that inputs are non-null. @param domainId the domain id @param domainCategoryId the domain category id @return the Set of event ids	Given the pricing context, the policy chart level and the date, this method evaluates the appropriate reward policy chart and returns it. Ensure that all inputs are non-null. No null-checks are performed. @param pricingContext the pricing context @param policyChartLevel the policy chart level @param evaluationDate the date for which to evaluate @return evaluated policy chart of type {@link PriceOfferingElement}
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllEventIds	Returns a list containing ids of all the events present in the repository, with the given domain and domain category. Ensure that inputs are non-null. @param domainId the domain id @param domainCategoryId the domain category id @return the Set of event ids	This method returns all events defined for given domain and domain category. @param domainId @param domainCategoryId @return @author GauravGh @since R2.3
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchIBEnabledTDProducts		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchIBEnabledTDProducts		@see com.ofss.fc.app.pricing.service.contract.WrappedContractApplicationService#getIBRegularInterestRatesFromPPC(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String)
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchIBEnabledCASAPr		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchIBEnabledCASAPr		@see com.ofss.fc.app.pricing.service.contract.WrappedContractApplicationService#getIBRegularInterestRatesFromPPC(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String)
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductDetails		Evaluates the policy chart for regular interest indices. On its evaluation, an <a href="#">{@link EvaluatedProductWiseInterestRatesInquiryResponse}</a> is obtained, detailing the regular interest index offering. This method will not persist a pricing recommendation. This method takes a array of offer id's and gives rates for offer which rates are defined in PPC.s <p> In this service we call <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)}</a> internally to acquire a <a href="#">{@link PricingTicketDTO}</a> . </p> @param sessionContext @param pricingContextDTO @param offerIds @param asOnDate @return @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchEventAmortizationLinkage	This method fetches event amortization linkage from the day 0 set-up. @param eventId @return <a href="#">{@link EventAmortizationLinkageDTO}</a>	This method fetches event amortization linkage from the day 0 set-up. @param eventId @return <a href="#">{@link EventAmortizationLinkageDTO}</a>
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchEventAmortizationLinkage	This method fetches event amortization linkage from the day 0 set-up. @param eventId @return <a href="#">{@link EventAmortizationLinkageDTO}</a>	
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchEventAmortizationLinkage	This method fetches event amortization linkage from the day 0 set-up. @param eventId @return <a href="#">{@link EventAmortizationLinkageDTO}</a>	This method will get all the price definitions which are attached to action <a href="#">{@link Action}</a> and outcomes <a href="#">{@link Outcome}</a> @param statement

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllProductGroupsForProductClass		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllProductGroupsForProductClass		This method processes particular addition done in Price Policy Chart @param sessionContext the session context @param pricePolicyChartAddition {@link PricePolicyChartAddition} to be processed @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllProductsForProductGroup		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllProductsForProductGroup		This method fetches the list of accounts that are to be repriced due to the addition of new PPC. It calls the respective account module with the list of product Offer Ids. @param sessionContext @param domainCategoryId @param groupList @return list of accounts @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllOffersForBundle		This method includes events or stages without benefits in PriceBenefitChartDTO. It fetches fee details for all the offer's included in the bundle or campaign and includes those events or stages which are not included in the Benefit Chart. @param priceBenefitChartDTO @return @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductClassForOffer		Helper method to fetch product class for given offer @param offerId Offer Id of the Offer @return Product Class of the Offer if Offer exists, null otherwise

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductClassForOffer		This method is used to get the fee details defined in the PPC under the offerId @param sessionContext @param offerId @return @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductClassForOffer		This method includes events or stages without benefits in PriceBenefitChartDTO. It fetches fee details for all the offer's included in the bundle or campaign and includes those events or stages which are not included in the Benefit Chart. @param priceBenefitChartDTO @return @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllowedCurrenciesForOffer		This method includes events or stages without benefits in PriceBenefitChartDTO. It fetches fee details for all the offer's included in the bundle or campaign and includes those events or stages which are not included in the Benefit Chart. @param priceBenefitChartDTO @return @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchProductGroupCodeForOffer		It makes a call to PMU module to fetch product group code respective of the offer Id. @param offerId @return productGroupCode @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllOfferForProduct		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchAllOfferForProduct		This method fetches the list of accounts that are to be repriced due to the addition of new PPC. It calls the respective account module with the list of product Offer Ids. @param sessionContext @param domainCategoryId @param groupList @return list of accounts @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	validateCurrencyForOffer		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	validateCurrencyForOffer		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	validateCurrencyForOffer		This method is used to fetch offers with permissible currencies for benefit. @param sessionContext @return BenefitOfferDetailsInquiryResponse @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchDDABankPolicy		
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchDDABankPolicy		This method fetches {@link DDABankPolicyDTO} for bank code present in session context @return {@link DDABankPolicyDTO} for bank code present in session context @throws FatalException
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchLoanBankPolicy		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingProductAdapter	fetchLoanBankPolicy		This method fetches the back dated for the respective bankcode. @return backdated duration @throws FatalException
IInsuranceLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceLCMAAdapter	getCreditFacilityDTOs		Fetches the facility type of linkages for the insurance id given and gets the Facility details in a CreditFacilityDTO[] @param lcmAdapter @param insuranceId @return @throws FatalException
IInsuranceLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceLCMAAdapter	fetchBorrowingEntity		
IInsuranceLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceLCMAAdapter	fetchBorrowingEntity		
IInsuranceLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceLCMAAdapter	fetchBorrowingEntity		Fetches borrowing entity details for BEs in CreditFacilityDTO[] @param lcmAdapter @param creditFacilityDTOArray @return @throws FatalException
IInsuranceLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceLCMAAdapter	fetchBorrowingEntity		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		Fetches the facility type of linkages for the insurance id given and gets the Facility details in a CreditFacilityDTO[] @param lcmAdapter @param insuranceId @return @throws FatalException
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchProposedFacility		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		Fetches the facility type of linkages for the insurance id given and gets the Facility details in a CreditFacilityDTO[] @param lcmAdapter @param insuranceId @return @throws FatalException
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchFacility		
IInsuranceLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLCMAadapter	fetchInsuranceLinkageforFacility		
IAgentDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This service will return AccountSummaryDTO for all the account under facilityId provided in input. @param facilityId :facility id of which all child Account will be returned back @return AccountSummaryResponse
IAgentDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is responsible for fetching the account based on the AccountType. If account type is not present throw business exception. @param accountId @param accountType @return AccountDTO @throws FatalException
ILoanProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanProcessAdapter	initiateLoanAccountRetentionProcess		<p>This method is responsible for raising Account Closure Request for a Loan Account.  Once the request is raised, Loan Account status is changed to "Marked for Closure". </p> @param sessionContext @param loanAccountClosureRequestDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProcessAdapter	initiateLoanAccountRetentionProcess		@param loanAccountClosureRequestDTO @return @throws FatalException
ILoanProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProcessAdapter	initiateHardshipReliefProcess		
ILoanProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanProcessAdapter	initiateHardshipReliefProcess		
IAccountingEntryProductCodeExtractionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingEntryProductCodeExtractionAdapter	extractProductCode		
IPricingWaiverAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingWaiverAdapter	inquireCustomWaiverDetails	This method will inquire details of custom waiver applicable for particular price and event. @param pricingContextDTO @param waiverDetailsDTO @return updated {@link WaiverDetailsDTO} with amount to be waived.	This method processes auto refund of fee (waiver) charged to an account and calls consulting to determine amount to be waived. It will {@link ProcessedBasePrice} with amount to be waived and makes call to next handler.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingWaiverAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingWaiverAdapter	inquireCustomWaiverDetails	This method will inquire details of custom waiver applicable for particular price and event. @param pricingContextDTO @param waiverDetailsDTO @return updated {@link WaiverDetailsDTO} with amount to be waived.	@see com.ofss.fc.app.adapter.AdapterFactory#getAdapter(java.lang.String, com.ofss.fc.datatype.NameValuePair[])
IAccountCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountCardAdapter	fetchCreditCardAccountDetails	Fetch Credit Card Account Details @param sessionContext @param accountId @return @throws FatalException	
IAccountCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountCardAdapter	fetchCreditCardAccountDetails	Fetch Credit Card Account Details @param sessionContext @param accountId @return @throws FatalException	The following method is used to fetch account related information using the account type. The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	creditFunds	This method will deposit the funds in clearing bridge ledger so that TD can operationally use the same. This method will be called for direct credits like incoming payments. It will park the funds in the clearing bridge ledger and expect the TD module to pick the funds from there for its liability gl	This method will deposit the funds in clearing bridge ledger so that TD can operationally use the same. This method will be called for direct credits like incoming payments. It will park the funds in the clearing bridge ledger and expect the TD module to pick the funds from there for its liability gl

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	creditFunds	This method will deposit the funds in clearing bridge ledger so that TD can operationally use the same. This method will be called for direct credits like incoming payments. It will park the funds in the clearing bridge ledger and expect the TD module to pick the funds from there for its liability gl	
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	markStatusAsRegular	This method will mark the TD as Regular state from the FundsAwaited state. It will then forward the accounting events from the TD module for raising the accounting events	This method will mark the TD as Regular state from the FundsAwaited state. It will then forward the accounting events from the TD module for raising the accounting events
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	markStatusAsRegular	This method will mark the TD as Regular state from the FundsAwaited state. It will then forward the accounting events from the TD module for raising the accounting events	
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	markStatusAsClosed	@param accountTransactionDTO @return	
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	markStatusAsClosed	@param accountTransactionDTO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsTDAdapter	reverseCreditFunds	@param accountTransactionDTO @param isPendingFunds @return @throws FatalException	
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchAlternateAccounts		This method would be used to fetch the alternate accounts for the given account.  It will also fetch the account for which it is alternate accounts
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchAlternateAccounts		This method is used to fetch the alternate account for the given account id. @param accountId @return
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchAlternateAccounts		This method will fetch what are the related accounts for provided account.The caller has to derive the summarise list of the information.The caller can be any account i.e. CASA,TD.Then caller has to take care of implementing class what is the condition of calling. @param applicationContext @param account
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchOffsetBalanceAccounts		This method is used to fetch the offset accounts for the given account id
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchOffsetBalanceAccounts		This method will fetch what are the related accounts for provided account.The caller has to derive the summarise list of the information.The caller can be any account i.e. CASA,TD.Then caller has to take care of implementing class what is the condition of calling. @param applicationContext @param account



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	getTaskModuleType		This method is used to fetch the offset accounts for the given account id
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchInstructionsBasedAccounts		This method is used to fetch the instructions based account for the given account.
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchInstructionsBasedAccounts		This method will fetch what are the related accounts for provided account.The caller has to derive the summarise list of the information.The caller can be any account i.e. CASA,TD.Then caller has to take care of implementing class what is the condition of calling. @param applicationContext @param account
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchSharedLimitsAccounts		This method will fetch the shared limit accounts for the given account.
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	fetchSharedLimitsAccounts		This method will fetch what are the related accounts for provided account.The caller has to derive the summarise list of the information.The caller can be any account i.e. CASA,TD.Then caller has to take care of implemenating class what is the condition of calling. @param applicationContext @param account
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	build		This method is used to build the casa response during the every action invocation.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	build		This method will call casa to get the simulated balance
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	saveClosureSimulation		This method is used to save the casa type closure simulation details
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	saveClosureSimulation		This will persist the @DDAClosureSimulation. The ID will be generated there with UUID. Then the closure record persisted will have information of posting date on which closure is requested. No of days after which closure is requested and account id for which closure is requested. It also consists of array of EffectiveBalances @link EffectiveBalances which will be persisted @param accountId @param noOfDays
ISimulationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationDDAAdapter	saveClosureSimulation		This is the method which performs simulation of data for the processing. This method calls DDASimulationProcessor#executeTestCase after assembling the data in DDALifeCycleInformationDTO @link DDALifeCycleInformationDTO. This method then assembles the test case required for lifeCycleExecution @link AbstractChainedLifeCycleAction. Then the execution starts. This method gets the optional input additionalDebit after calling @link DDALoanAdapter#deriveScheduleForPeriodicDebit and additionalCredit after calling @link DDATDAAdapter#deriveScheduleForPeriodicCredit. Now that is assembled into DDALifeCycleInformationDTO and passed for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				processing. @param accountId @param noOfDays
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	calculateScheduleInterest	@param applicationContext @param accountScheduleAttributes @param period @param balanceRecordDTO @param interestRates @return @throws FatalException	This method is called during Interest Simulation. 1>This method fetches the Loan Account Information by loading the information of the account Id. 2>Simulation is always done for Regular Interest Division.Hence InterestDivision is hard coded as Regular 3>Method populateInterestEventRequestDTO is called 4>LoanBalanceInformationDivisionMap is prepared . For preparing this map fetchLoanBalanceInformationDTO method is called. 4>DateRange is set to the loanInterestEventDTO. 5>RateChartDetails map is prepared .For preparing this map prepareRateChartDTO method is called. 6>CalculateScheduleInterest method of LoanInerestApdater is called . @param appContext @param accountParameters @param dateRange @param scheduleInterestBase @param stage @return
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	calculateScheduleInterest	@param applicationContext @param accountScheduleAttributes @param period @param balanceRecordDTO @param interestRates @return @throws FatalException	
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	inquireScheduleInterest	@param applicationContext @param loansInterestSimulationRequest @return @throws FatalException	@param accountParameters @param dateRange @param scheduleInterestBase @param stage @param interestRates @return @throws FatalException
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	inquireScheduleInterest	@param applicationContext @param loansInterestSimulationRequest @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	inquireUnbilledInterest	@param accountId @param dueDate @param isClosure @return	@param scheduleId @param valueDate @return
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	inquireBackdatedAccruedRegularInterest	This method inquires the backdated accrued regular interest. @param scheduleId @param fromDate @param toDate @param isHardshipRelief @return	This method inquires the backdated accrued regular interest. @param scheduleId @param fromDate @param toDate @param isHardshipRelief @return
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	inquireLastChargedInterest	This method returns the last charged interest amount. @param scheduleId @return @throws FatalException	@param scheduleId @return @throws FatalException
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	updateNextInterestCapDate	This method is used to update next capitalization date in case if the interest cap/charge frequency of the schedule is getting updated This is required because Interest module accrues interest till next cap/charge date. If the frequency of the interest charging/cap changes we need to notify interest regarding the same, so the accrual will be in sync with the interest charge/cap frequency of the schedule	This method will be called at the time backdated reschedule/ restructure event as well as replay events. It takes inputs like account id, event date, event based history number & new schedule code (in case of backdated restructure). This method calls - A] PrincipalSchedulePeriodHistoryRepository for schedule configuration at the time of reschedule either based on event based history number or event date. B] LoanBalanceAggregateRepository to fetch schedule principal balance at the time reschedule or replay event. C] LoanScheduleTypeRepository to fetch schedule type details in case of backdated restructure where new schedule code is supplied by the framework. D] doBackdatedReschedule() method to generate summary schedule E] generateBackdatedRecordsOnReschedule() method to generate blown up schedule. F] saveBackdatedSchedule to save the backdated

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				schedule in the main table. @param accountId @param eventDate @param eventNumber @param newScheduleCode
IScheduleInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\schedule\IScheduleInterestAdapter	updateNextInterestCapDate	This method is used to update next capitalization date in case if the interest cap/charge frequency of the schedule is getting updated This is required because Interest module accrues interest till next cap/charge date. If the frequency of the interest charging/cap changes we need to notify interest regarding the same, so the accrual will be in sync with the interest charge/cap frequency of the schedule	This method is used to update next capitalization date in case if the interest cap/charge frequency of the schedule is getting updated This is required because Interest module accrues interest till next cap/charge date. If the frequency of the interest charging/cap changes we need to notify interest regarding the same, so the accrual will be in sync with the interest charge/cap frequency of the schedule
ILCMSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ILCMSecurityAdapter	fetchCreditExposureDetails		Method to get the Credit Exposure Details from LCM module (CreditExposureApplicationService.fetchCreditExposureDetails()).
ILCMSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ILCMSecurityAdapter	fetchCreditExposureDetails		
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	build		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	build		The method will build simulation response for hardship relief simulation. The response will contain the indicative interest benefit which customer will get if given hardship relief is applied to the account.
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	build		
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	build		
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	build		@param lifeCycleInformationDTO @return
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	applyHardshipRelief	The method is responsible to apply hardship relief to account. @param loanHardshipReliefDetailsDTO @throws FatalException	The method is responsible to apply hardship relief to account.
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildHardshipReliefResponse	The method will build simulation response for hardship relief simulation. The response will contain the indicative interest benefit which customer will get if given hardship relief is applied to the account. @param	The method will build simulation response for hardship relief simulation. The response will contain the indicative interest benefit which customer will get if given hardship relief is applied to the account.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			lifeCycleInformationDTO @return	
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildHardshipReliefResponse	The method will build simulation response for hardship relief simulation. The response will contain the indicative interest benefit which customer will get if given hardship relief is applied to the account. @param lifeCycleInformationDTO @return	@param lifeCycleInformationDTO @return
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	calculateInterestAmountWithoutHardshipRelief		The method is responsible to apply hardship relief to account.
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildRolloverResponse	@param lifeCycleInformationDTO @return LifeCycleResponseDTO	
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildRolloverResponse	@param lifeCycleInformationDTO @return LifeCycleResponseDTO	@param lifeCycleInformationDTO @return
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildFutureDatedSplitResponse	@param lifeCycleInformationDTO @return LifeCycleResponseDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	buildFutureDatedSplitResponse	@param lifeCycleInformationDTO @return LifeCycleResponseDTO	@param lifeCycleInformationDTO @return
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	addRolloverInstruction		
ISimulationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\simulation\ISimulationLoanAdapter	addRolloverInstruction		The method is responsible to add rollover instruction to account.
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	processActivityEvents	Used to trigger processing of events which are generated as part of activity  in application.	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	This method logs/registers the activity log DTO in the Alerts table. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	This method logs/registers the activity log DTO in the Alerts table for batch processing. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	This method logs/registers the activity log DTO in the Alerts table for batch processing. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	This method logs/registers the activity log DTO in the Alerts table for batch processing. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerEventsForActivity	Used to register events for the activity in application.	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	processTransactionalEvents	Used to trigger processing of transactional events which are generated as part of activity @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	initiateEventPoller		
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	initiateEventPoller		Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller , delayed transaction poller , FTP webservice poller. Please add more in this method Additionally we also have an extension for consulting pollers @param sessionContext
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	parkActivityLog		
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	parkActivityLog		This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of events  generated as part of the activity registration happens only after the activity completes  successfully. The processing of events generated happens asynchronously in separate thread  and has it own processing lifecycle. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context   information such as user id, branch, branch code, channel etc. @fcb.param MI,String,activityId, , Represents the activity that has occurred for which the   log data is being registered and configured events will be generated. Example of such activity  could be execution of a business service like cheque deposit, loan installment

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>payment etc.&lt;br&gt; @fcb.param  MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	parkActivityLog		<p>Please see the method document at {@link this#registerActivityAndGenerateEvent(SessionContext, ActivityEventKeyDTO, Date, ActivityLog)} If alert is maintain customer subscribed and no subscriber is attached then activity data is logged in to the system but alert will not be raised. @param bankCode @param activityEvent @param eventProcessingDate @param activityLog @param status @param adapter @param response @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	overwriteActivityLog		<p>This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle.</p> <p>@fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,String,activityId, , Represents the activity that has occurred for which the &lt;br&gt; log data is being registered and configured events will be generated. Example of such activity&lt;br&gt; could be execution of a business service like cheque deposit, loan installment payment etc.&lt;br&gt; @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	overwriteActivityLog		Please see the method document at (@link this#registerActivityAndGenerateEvent(SessionContext, ActivityEventKeyDTO, Date, ActivityLog)) If alert is maintain customer subscribed and no subscriber is attached then activity data is logged in to the system but alert will not be raised. @param bankCode @param activityEvent @param eventProcessingDate @param activityLog @param status @param adapter @param response @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchActivityLog		
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	clearActivityLog		This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of events  generated as part of the activity registration happens only after the activity completes  successfully. The processing of events generated happens asynchronously in separate thread  and has it own processing lifecycle. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context   information such as user id, branch, branch code, channel etc. @fcb.param MI,String,activityId, , Represents the activity that has occurred for which the   log data is being registered and configured events will be generated. Example of such activity  could be execution of a business service like cheque deposit, loan installment payment etc.  @fcb.param MI,ActivityLog,activityLog, , Data associated with the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	clearActivityLog		<p>This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a seperate parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt; @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	clearActivityLog		<p>This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a separate parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,EventContext,eventContext, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	clearActivityLog		<p>Required for performance improvement in batch mode calls so that this method does not flush.This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,Strin ,bankcode, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a seperate</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt; @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	clearActivityLog		<p>Please see the method document at {@link this#registerActivityAndGenerateEvent(SessionContext, ActivityEventKeyDTO, Date, ActivityLog)} If alert is maintain customer subscribed and no subscriber is attached then activity data is logged in to the system but alert will not be raised. @param bankCode @param activityEvent @param eventProcessingDate @param activityLog @param status @param adapter @param response @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	executeRule	Used to execute rule associated with an alert  @throws FatalException	This method executes a rule and returns the result of execution. It checks  if the rule engine which would execute the rule is internal or external one.  For internal rule execution, it creates an instance of rule class configured  reflectively and invokes the { <a href="#">@link</a> #com.ofss.fc.app.integration.ep.event.IRuleHandler.execute(IActivityLog)} method.  For an external rule engine, the rule execution call is delegated to the rule  engine. Any exception or error raised during rule execution is logged and a  failure result is returned to the calling program. @fcb.param MI,DecisionAgent,agent, , This class holds the instance of rule   configured and the decision of rule execution. @fcb.param MI,IActivityLog,activityLog, , This is the activity log which was  generated while registering the activity. @return response A boolean indicating success or failure of rule execution. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	executeRule	Used to execute rule associated with an alert  @throws FatalException	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the Alert must be send or not. @param String, ruleId @param String, domainCategoryId @param String, domainId @param com .ofss.fc.dataType.Date, rule effective date @param IActivityLog, activityLog @param Set, ruleDataSources IRuleDataSource @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	executeRule	Used to execute rule associated with an alert  @throws FatalException	<p>This service is used to validate the applicants applying for different variants under a submission.&lt;br&gt; These options are made available based on rules setup within the system. These options are granted only&lt;br&gt; when the rules return a positive outcome.&lt;br&gt; Existing applicants that are employees can apply for employee benefits.&lt;br&gt; For Refinancing of loans, fast refinance option is available.&lt;br&gt; For applicants with employment restrictions, option for low doc is available.&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#APPLICANT_SUBMISSION_VALIDATION_FAILED} : FC_OR_DE_0088 &lt;br&gt; This exception occurs applicants applying for variants are not applicable. Business validation error is thrown</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>and variants are not selected. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#EMPLOYEE_BENEFITS_NOT_APPLICABLE} : FC_OR_DE_0080 &lt;br&gt; This exception occurs when employee benefits flag is checked for a Product Group &lt;br&gt; but Employee benefit Applicability Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#LOW_DOC_NOT_APPLICABLE} : FC_OR_DE_0079 &lt;br&gt; This exception occurs when Low Doc Applicable flag is checked for a Product Group &lt;br&gt; but Low doc Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_AUTO_REFI_NOT_APPLICABLE} : FC_OR_AR_0001 &lt;br&gt; This exception occurs when Fast Refi flag is checked for a Product Group &lt;br&gt; but Fast Refi Rule Execution returns Not Applicable.&lt;br&gt;</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO in the Alerts table for batch processing. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This represents the updatePartyWarningIndicator method of the entity. This method gets called whenever warning indicator is linked or delinked from party. It also updates the warning indicator changes on party to other modules where it has an impact. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,{@link PartyWarningIndicatorDTO},partyWarningIndicatorDTO PartyWarningIndicatorDTO contains the following: partyId: The warning indicators will be set for this party id. partyType: Mandatory Input This will indicate if the party is individual, organization or banking entity. warningIndicatorId: The type of warning Indicator. isApplicable: It indicates whether the warning indicator is set or not. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	<p>This method will register the activity request with event processing. </p> @param activityId @param eventId @param partyId @param bundleName @param processDate @param accountNumber @param feeAmount @param currencyCode
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method will raise the alert for system and user defined status account status.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO in the Alerts table. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO in the Alerts table. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,PartyType,partyType, holds the type of party - Individual, Bank etc. @fcb.param MI,String,eventId, holds the event Id. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	@param aSessionContext @param aEventId @param aActivityId @param aActivityLog @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId the activity id against which alerts needs to be sent @param eventId the event id against which alerts needs to be sent @param activityLog the activity log DTO containing details of the account no,txn amount to be used as parameters in alerts @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is designed to raise a request for Account Closure. For further details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosureApplicationService#createClosureRequest(SessionContext, DDAClosureRequestDTO)}
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method triggers alert for updating notifications. @param applicationContext @param partyId @param source
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service {@link ActivityRegistrationApplicationService#registerActivityAndGenerateEventwithoutInteraction(String, ActivityEventKeyDTO, Date, ActivityLog)} which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method register an activity and events for all associated events in table. Through this method Events will be registered or logged in event processing module for further processing of Events and send respective alerts. @param applicationContext @param activityLog @param applicationServiceId @param eventId
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	@param aSessionContext @param aEventId @param aActivityId @param aActivityLog @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to persist the activity. @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method register an activity and events for all associated events in table. @param applicationContext @param activityLog @param applicationServiceId @param eventId
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventId @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,PartyType,partyType, holds the type of party - Individual, Bank etc. @fcb.param MI,String,eventId, holds the event Id. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,String,partyName, holds the Party Name. @fcb.param MI,String,eventId, holds the event Id. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,String,partyName, holds the Party Name. @fcb.param MI,String,eventId, holds the event Id. @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,String,partyName, holds the Party Name. @fcb.param MI,String,partyType, holds the Party type. @fcb.param MI,String,eventId, holds the event Id. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method logs/registers the activity log DTO @fcb.param MI,String,activityId, The ActivityId for which we need to log the data for the further processing of alerts. @fcb.param MI,Object,object, holds the new data, sent as DTO's. @fcb.param MI,SessionContext,sessionContext, holds the session data, used for creating the ApplicationContext. @fcb.param MI,String,partyName, holds the Party Name. @fcb.param MI,String,eventId, holds the event Id. @fcb.param MI,String[], updatedLocationNames, holds Location names in case of KYC location changes @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to register activity(flx_ep_act_log_b) and events(flx_ep_evt_log_b). @param sessionContext @param activityLog @param activityId @param eventId
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method triggers alert for updating notifications. @param applicationContext @param partyId @param source
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to send alerts to all parties linked with specific party as owner , administrator or others etc. @param applicationContext @param originationActivityDTO @param sessionContext @param activityLog @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to close Deposit. @param accountId Account id to which deposit belongs @param depositId Identifier of Deposit to be closed @return {@link TransactionStatus} containing result of close operation @see com.ofss.fc.app.td.service.deposit.IDepositApplicationService#closeDeposit(com.ofss.fc.app.context.SessionContext, java.lang.String, int)
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of events  generated as part of the activity registration happens only after the activity completes  successfully. The processing of events generated happens asynchronously in separate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a seperate parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt; @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of events  generated as part of the activity registration happens only after the activity completes  successfully. The processing of events

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a seperate parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt; @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,EventContext,eventContext, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	Required for performance improvement in batch mode calls so that this method does not flush.This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>events generated as part of the activity registration happens only after the activity completes successfully. The processing of events generated happens asynchronously in separate thread and has its own processing lifecycle. This method facilitates future dated event action processing. @fcb.param MI,Strin ,bankcode, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred. The log data registered would be for the activity and the events that is sent along with the will be generated with a specific processing date passed as a separate parameter. Example of Example such activity could be execution of a business service like cheque deposit, loan installment payment etc. @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for event processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is stored in the database for future retrieval and processing of events associated with the activity registered. @return response An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEvent	Used to register an Activity with an associated Event @param activityID @param eventID @param eventProcessingDate @param activityLog @return @throws FatalException	Internal method used to load the activity event by making the repository call.
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroups		This represents the fetchApprovalGroups method of the entity.  It makes a call to SOA server and retrieves all the approval groups.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @return {@link com.ofss.fc.app.sms.dto.user.ApproverGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroups		Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroupMembers	Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException	Gets the banker approver details. @param rmt @param sourcePartyId @return @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroupMembers	Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException	Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroupMemberDispatchDetail	Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException	Gets the banker approver details. @param rmt @param sourcePartyId @return @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchApprovalGroupMemberDispatchDetail	Fetch Approval Groups from SMS @param sessionContext @return @throws FatalException	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	deleteActivityEventLog		This method is used to delete activity log data. @param sessionContext @param activityId @param partyAcctType @param applicationId @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	deleteActivityEventLog		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	deleteActivityEventLog		This method gets called when the user chooses to come back to Structure solution stage from any of the below stages for any lending application   <ul> <li>Mortgage Insurance</li> <li>Create Offer</li> <li>Lending Account Opening</li> <li>Lending Settlement</li> </ul> Method resets the below data, if present   1. Cancel the LMI policy  2. Delete the auto refinance data  3. Delete offer data  Method also performs the state transition @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String, applicationId, , Indicates the application id of the application to reset. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception   {@link OriginationErrorConstants#SS_REVERT_NOT_VALID } : FC_OR_SS_0020  This error occurs when:  Structure solution is reverted when account has already been opened for one or more product groups.
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	deleteActivityEventLog		Internal method used for deleting data for given inputs by making repository call. @param sessionContext @param activityId @param partyAcctType @param applicationId @author VirendraO
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	This method is used to register an activity along with the data relevant for the activity  being registered. Additionally, the methods queries for events which are registered with  the activity being performed and generates them for processing. The processing of events  generated as part of the activity registration happens only after the activity completes  successfully. The processing of events generated happens asynchronously in separate thread  and has it own processing lifecycle. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context   information such as user id, branch, branch code, channel etc. @fcb.param MI,String,activityId, , Represents the activity that has occurred for which the   log data is being registered and configured events will be generated. Example of such activity  could be execution of a business service like cheque deposit, loan installment payment etc.  @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is   stored in the database for future retrieval and processing of events associated with the activity  registered. @return response An instance of the transaction status which contains the required information  about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	This method is a helper method used for registering activity in bulk and generate the events configured against the activity. This method simply delegates the call to <code>registerActivityAndGenerateConfiguredEvents(SessionContext, String, ActivityLog)</code> in a loop. In this case, every single activity log is registered against the activity in a separate transaction scope and committed separately. @deprecated
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	This method is used to register an activity along with the data relevant for the activity being registered. Additionally, the methods queries for events which are registered with the activity being performed and generates them for processing. The processing of events generated as part of the activity registration happens only after the activity completes successfully. The processing of events generated happens asynchronously in separate thread and has its own processing lifecycle. This method facilitates future dated event action processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred. The log data registered would be for the activity and the events that is sent along with the will be generated with a specific processing date passed as a separate parameter. Example of Example such activity could be execution of a business service like cheque deposit, loan installment payment etc. @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for event

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	<p>This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,SessionContext,sessionContext, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a separate parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,EventContext,eventContext, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	populateMultiEntityAttributes	Populate Multi Entity Attributes of ActivityLog @param activityLog @return @throws FatalException	<p>Required for performance improvement in batch mode calls so that this method does not flush.This method is used to register an activity along with the data relevant for the activity&lt;br&gt; being registered. Additionally, the methods queries for events which are registered with&lt;br&gt; the activity being performed and generates them for processing. The processing of events&lt;br&gt; generated as part of the activity registration happens only after the activity completes&lt;br&gt; successfully. The processing of events generated happens asynchronously in separate thread&lt;br&gt; and has it own processing lifecycle. This method facilitates future dated event action&lt;br&gt; processing. @fcb.param MI,Strin ,bankcode, , Session Context which must contain the session context &lt;br&gt; information such as user id, branch, branch code, channel etc. @fcb.param MI,ActivityEventKeyDTO,activityEvent, , Represents the activity that has occurred.&lt;br&gt; The log data registered would be for the activity and the events that is sent along with the&lt;br&gt; will be generated with a specific processing date passed as a seperate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>parameter. Example of &lt;br&gt; Example such activity could be execution of a business service like cheque deposit, loan &lt;br&gt; installment payment etc.&lt;br&gt; @fcb.param MI,Date,eventProcessingDate, , Date on which the event should be picked up for &lt;br&gt; event processing &amp; action handling. @fcb.param MI,ActivityLog,activityLog, , Data associated with the activity performed. This data is &lt;br&gt; stored in the database for future retrieval and processing of events associated with the activity&lt;br&gt; registered. @return response An instance of the transaction status which contains the required information&lt;br&gt; about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @see com.ofss.fc.app.ep.service.event.EventHandler</p>
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	<p>Used to register an Activity with an associated Event.&lt;br&gt;This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close.  @param applicationContext  @param activityID @param eventID  @param eventProcessingDate  @param logObject @return activityDataId @throws FatalException</p>	<p>This method logs/registers the activity log DTO in the Alerts table for batch processing. @fcb.param MI,String,activityId, , The ActivityId for which we need to log the data for the further processing of alerts @fcb.param MI,IActivityLog,activityLog, ,holds the data elements which needs to be logged. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException	<p>This method will register the activity request with event processing. </p> @param activityId @param eventId @param partyId @param bundleName @param processDate @param accountNumber @param feeAmount @param currencyCode
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId the activity id against which alerts needs to be sent @param eventId the event id against which alerts needs to be sent @param activityLog the activity log DTO containing details of the account no,txn amount to be used as parameters in alerts @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	registerActivityAndGenerateEventInBatch	Used to register an Activity with an associated Event. This method has been written as performance fix to avoid flush call in batch as well as online. It will call service ActivityRegistrationApplicationService#registerActivityAndGenerateEvent(String, ActivityEventKeyDTO, Date, ActivityLog) which does not use interaction begin and close. @param applicationContext @param activityID @param eventID @param eventProcessingDate @param logObject @return activityDataId @throws FatalException	This method registers activity and generates event for given activity id and event id @param activityId @param eventId @param activityLog @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchBulletinRecipients		This method process the subscribes alerts. @see com.ofss.fc.domain.ep.service.action.ActionHandlerService#processAction(com.ofss.fc.entity.integration.ep.action.IAction, com.ofss.fc.framework.domain.ep.dto.ActivityData, com.ofss.fc.entity.integration.ep.action.subscriber.ISubscriber, com.ofss.fc.entity.integration.ep.action.IActionTemplate)
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	fetchBulletinRecipients		Gets the Bulletin board data. @param rmt @param partyId @param adapterFactory @return @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	getAccountDetails	Used to fetch the account details on the basis of account details provided. @param accountId @return AccountDetailsResponse @throws FatalException	This methods sets the multi-entity attributes for the alert processing.  If account passed in the request for the processing then it load the account on the basis of type of account. @param activityData @param subscription @return @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	getAccountDetails	Used to fetch the account details on the basis of account details provided. @param accountId @return AccountDetailsResponse @throws FatalException	This method fetches an instance of the action handler service based on the  action being processed and delegates the call to handler for further processing. @fcb.param MI,ISubscriber,subscriber, , Subscriber whose subscriber actions are being processed is passed as an argument. @fcb.param MI,ISubscribedAction[],subscribedEventActions, , Actions of events which have been subscribed for. @fcb.param MI,ActivityData,activityData, , Data registered along with the  activity registered.  @throws FatalException
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	getAccountDetails	Used to fetch the account details on the basis of account details provided. @param accountId @return AccountDetailsResponse @throws FatalException	This method performs action for the event. This method checks if the action to be performed allows for subscription. If subscription is allowed, the method delegates the call to the domain service exposed by @see com.ofss.fc.domain.ep.service.subscription.ActionSubscriptionProcessorService namely, processActionSubscriptions   In case subscription is not allowed for the action being processed the method internally delegates the call to {@link #processUnsubscribedEventAction(IActivityEventAction, ActivityData)}   @fcb.param MI,IActivityEventAction,eventAction, , An action associated with the event. @fcb.param MI,ActivityData,activityData, , Data registered along with the activity to facilitate rule evaluation.  @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	getAlertsDefaultConfiguration	This method set the default data from multientities alert configuration. @param businessUnitCode @param marketEntityCode @param alertType @param destinationType @return	
IEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ep\IEventProcessingAdapter	getAlertsDefaultConfiguration	This method set the default data from multientities alert configuration. @param businessUnitCode @param marketEntityCode @param alertType @param destinationType @return	This method is use to fetch the alert configuration details from multi-entity. @param businessUnitCode @param marketEntityCode @param alertType @param destinationType @return MultiEntityAlertConfInquiryResponse
IPricingLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchLoanAccountsForProductOffer	This method will call the Loan service to fetch the list of Accounts opened under the given productOfferId. @param sessionContext @param productOfferId @return list of accounts @throws FatalException	
IPricingLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchLoanAccountsForProductOffer	This method will call the Loan service to fetch the list of Accounts opened under the given productOfferId. @param sessionContext @param productOfferId @return list of accounts @throws FatalException	This method fetches the list of accounts that are to be repriced due to the addition of new PPC. It calls the respective account module with the list of product Offer Ids. @param sessionContext @param domainCategoryId @param groupList @return list of accounts @throws FatalException
IPricingLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchLoanAccountsForDate		This method will fetch all the loan account for a date. @param date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	getFirstDisbursementDate		This method will derive the applicablePeriodEndDate on the basis of trackingStartDate for anniversary charge attribute.
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	getFirstDisbursementDate		This method is used to prepare stage and re-price date wise map for given account for index re-price @param domainCategoryId domain category id @param effectiveDate Effective date of index @param repriceStageDetailsSet Set of {@link RepriceStageDetails} containing stage details for index used @return stage and re-price date wise map for given account for index re-price @throws FatalException
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	getFirstDisbursementDate		@param indexCode @param effectiveDate @param arrangementsList @return @throws FatalException @deprecated Instead {@link RepricingDataCollectionService#searchAccountsForIndexReprice(String, Date, PriceType, String, boolean, List)} should be used. As indexes linked to account are stored in net rate entity itself.
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	getFirstDisbursementDate		Returns a list of type {@link PriceFeature}, containing the prices to be applied for all periodic events defined for the given arrangement/account number. The evaluation of the fee/price is done as per the effective date provided. The returned list will contain one element per periodic event. @param pricingContext the pricing context @param arrangementNumber the arrangement/account number @param evaluationDate the evaluation date @param currencyCodes the string array of currency codes @param periodFactInformationSet set of {@link PeriodFactInformation} @return an array of type {@link PriceFeature}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	getFirstDisbursementDate		Helper function that calls the {@link com.ofss.fc.domain.pricing.service.price.PriceComputationService PriceComputationService} for each of the {@link PriceOfferingElement PriceOfferingElements} within the set argument. It adds all the ProcessedBasePrices and applies the campaign variance to get the net amount, in the form of a MoneyBag containing all the currencies specified in the currencyCodes, which is then set to the totalPrice of the {@link PeriodicPriceFeature}. @param pricingContext the pricing context @param arrangementNumber the arrangement number @param eventCode the event code @param evaluationDate the evaluation date @param priceOfferings the set of {@link OfferingElement OfferingElements} @param currencyCodes the string array of currency codes @param periodicEventTracker the periodic event tracker containing list of {@link PeriodicFeeDate} @param computationDate the computation date @param pricingPlan the pricing plan @param periodFactInformationSet set of {@link PeriodFactInformation} @return the {@link PeriodicPriceFeature}
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchRepriceLoanAccountDetails		
IPricingLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchRepriceLoanAccountDetails		This method is used to fetch the loan account details for which first disbursement happened to the given date.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	fetchRepriceLoanAccountDetails		This method is used to fetch all accounts opened today @param domainCategoryId, dueDate @return List: accountOpenedToday @throws FatalException
IPricingLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingLoanAdapter	isFeeToBeNegotiated		Validates Event Price Variances @param domainId @param domainCategoryId @param accountId @param eventPriceVariances
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchConditionCode	This method fetches the condition details for a given condition Id. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, conditionId, , Indicates the identifier for the condition for which details are fetched @return ConditionCodeDTO , DTO containing details of condition code @throws FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateChargeId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateChargeId		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateCollateralId		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateCollateralId		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCovenantCode	This method fetches the covenant details for a given covenant Id. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, covenantCode, , Indicates the identifier for the covenant for which details are fetched @return CovenantDTO , DTO containing details of covenant @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This service returns a distinct list of collaterals for list of facility Ids passed @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionDTO submissionDTO, , Existing Submission Details @return CollateralDTO[], List of collaterals belonging to the facility list.
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This method will get the collterals for the application @param Application application, the application domain object @return CollateralDTO list - collaterals for that application @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This service returns a distinct list of collaterals for list of facility Ids passed @param sessionContext @param submissionDTO @return
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This method creates an array of {@link CollateralDTO} which contains information about the collateral which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link CollateralDTO} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, , {@link submissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This method creates an array of {@link CollateralDTO} which contains information about the collateral which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link CollateralDTO} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		This service will check and return if collateral perfection is required for any collateral in the submission @param sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param submissionId the id of the submission. @return the {@link CollateralPerfectionResponse} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralsForFacilityIdList		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	saveConditionLinkages		<p>This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:-</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Approve&lt;/li&gt;</li> <li>&lt;li&gt;Conditionally Approve&lt;/li&gt;</li> <li>&lt;li&gt;Refer&lt;/li&gt;</li> <li>&lt;li&gt;Refer Downsell&lt;/li&gt;</li> <li>&lt;li&gt;Decline&lt;/li&gt;</li> <li>&lt;li&gt;Continue&lt;/li&gt;</li> </ul> <p>Depending on the outcome of the call to decision interface, further processing of submission will take place. &lt;2.3&gt;, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline task. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response object of this call that contains the details of credit decision of submission and enitites within submission @exception&lt;br&gt; {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#SUBMISSION_NOT_FOUND) : FC_OR_DE_0055&lt;br&gt; This exception is thrown if submission is not found for the given submission id.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND) : FC_OR_DE_0075&lt;br&gt; This exception is thrown if submission history record is not found for the submission id which is being edited or amended.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_INT_SENSITIVITY_RATE_NOT_MNT) : FC_OR_CD_0026&lt;br&gt; This exception is thrown if interest sensitivity rate is not maintained for the lending product group and currency combination applied for in the submission.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_VALUATION_TYPE_NULL) : FC_OR_CD_0029&lt;br&gt; This exception is thrown for a collateral which requires valuation and if the no valuation type is recommended by the credit assessment engine.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	saveCovenantLinkages		<p>This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:-</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Approve&lt;/li&gt;</li> <li>&lt;li&gt;Conditionally Approve&lt;/li&gt;</li> <li>&lt;li&gt;Refer&lt;/li&gt;</li> <li>&lt;li&gt;Refer Downsell&lt;/li&gt;</li> <li>&lt;li&gt;Decline&lt;/li&gt;</li> <li>&lt;li&gt;Continue&lt;/li&gt;</li> </ul> <p>Depending on the outcome of the call to decision interface, further processing of submission will take place. &lt;2.3&gt;, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline task. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response object of this call that contains the details of credit decision of submission and enitites within submission @exception&lt;br&gt; {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#SUBMISSION_NOT_FOUND) : FC_OR_DE_0055&lt;br&gt; This exception is thrown if submission is not found for the given submission id.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND) : FC_OR_DE_0075&lt;br&gt; This exception is thrown if submission history record is not found for the submission id which is being edited or amended.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_INT_SENSITIVITY_RATE_NOT_MNT) : FC_OR_CD_0026&lt;br&gt; This exception is thrown if interest sensitivity rate is not maintained for the lending product group and currency combination applied for in the submission.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_VALUATION_TYPE_NULL) : FC_OR_CD_0029&lt;br&gt; This exception is thrown for a collateral which requires valuation and if the no valuation type is recommended by the credit assessment engine.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This service will populate the fact required for the Low Doc rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details @fcb.param MI, Map<String, CollateralDTO>, collateralMap, , Indicates the collateral details with collateral code as the key. Collaterals linked to the facility are fetched using this collateral code. @fcb.param MI,Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates the map containing applicant data with applicant serial number as the key. @return FactInformationDTO[], Returns a list of facts required for execution of Low Doc Rule. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method needs to be called if LRM is applicable at the product group level and will do the following: <ul style="list-style-type: none"> <li>&lt;li&gt;1.calculate LVR, execute rule to get submission type and provider type&lt;/li&gt;</li> <li>&lt;li&gt;2.Execute rule to get if DUA is applicable&lt;/li&gt;</li> </ul> @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001   This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001   This exception is thrown when we try to execute LRM rules for a Product Group for which LMI is not applicable. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011   This exception is thrown when Rule Id is not linked in Origination Preference for LRM Submission Type and LRM Provider Rules.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method will be invoked to evaluate LRMSubmissionType rule to get the Submission type for the LRM policy. This will be called in the pre-assessment stage. @param sessionContext @param submissionInputDTO @param productGroupSerialNo @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method needs to be called if LRM is applicable at the product group level and will do the following: <ul style="list-style-type: none"> <li>&lt;li&gt;1.Execute rule to check if LRM is required&lt;/li&gt;</li> <li>&lt;li&gt;2. If LRM is required,calculate LVR, execute rule to get submission type and provider type&lt;/li&gt;</li> </ul> @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MICreditFacilityDTO[] facilityDetailsDTOs , This contains the various collaterals linked to the Product Group. @return TransactionStatus
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR	This method will update the approved amount in the credit facility dto from lcm to be 0 and the product group linkage from origination to be 0 @param sessionContext @param submissionDecisionDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			value returned from LCM	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, , @link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:- <ul><li>Approve</li> <li>Conditionally Approve</li> <li>Refer</li> <li>Refer Downsell</li> <li>Decline</li> <li>Continue</li> </ul> Depending on the outcome of the call to decision interface, further processing of submission will take

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>place. &lt;2.3&gt;, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline task. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response object of this call that contains the details of credit decision of submission and enitites within submission @exception&lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055&lt;br&gt; This exception is thrown if submission is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND} : FC_OR_DE_0075&lt;br&gt; This exception is thrown if submission history record is not found for the submission id which is being edited or amended.&lt;br&gt; {@link OriginationErrorConstants#OR_INT_SENSITIVITY_RATE_NOT_MNT} : FC_OR_CD_0026&lt;br&gt; This exception is thrown if interest sensitivity rate is not maintained for the lending product group and currency combination applied for in the submission.&lt;br&gt; {@link OriginationErrorConstants#OR_VALUATION_TYPE_NULL} : FC_OR_CD_0029&lt;br&gt; This exception is thrown for a collateral which requires valuation and if the no valuation type is recommended by the credit</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				assessment engine.  @throws FatalException
IOrigination LCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method validates whether the LVR value has been changed from the last time value. If no change in LVR value then it return false otherwise returns true. @param sessionContext @param lendingPgLinkageDTO @param collaterals @param applicantDatas @param creditFacilityDTOs @return Boolean
IOrigination LCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This service is used to create new facilities or update existing facilities.  The following activities will be carried out in sequence.    a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the  facility. LCM internally checks if an existing borrowing entity exists for the parties passed and   returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity.   b) Update the borrowing entity identifier against the facility.   c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates   linkages to collaterals linked to the facility. Also, simulations that are attached to the facility  either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO,submissionInputDTO ,, Indicates the submission details containing the applicant and facility information. @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.   @throws FatalException @exception   {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076   This error occurs when:  The borrowing entity identifier for the group of applicants applying for a facility could not be generated. 
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method will inquire LVR for a given product group.This method is being called in both post and pre assessment. @param submissionInputDTO @param lendingPgLinkageDTO @param creditFacilityDTOs @return lvr
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This service is used to fetch LVR value for the facility post assessment, on Structure solution when before an account is created or amended to set it as input for simulation which will affect pricing.  Internally, this service will call inquireLVRForProductGroup which is common for both fetch LVR pre and post assessment @param submissionId @param lendingPgLinkageDTO @param facilities @return lvr

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method will update the LRM and LVR value preassessment
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using facility Id. This method will be called in the post-assessment stage. @param SessionContext - contains information such as user id, branch, branch code, channel etc. @param String facilityId, Identifier for the facility. @return BigDecimal - LVR value returned from LCM	This method will update the insurance amount for funding table whilst fetching funding table dto. In case of amend funding table , set the different in the amount. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This service will populate the fact required for the Low Doc rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details @fcb.param MI, Map<String, CollateralDTO>, collateralMap, , Indicates the collateral details with collateral code as the key. Collaterals linked to the facility are fetched using this collateral code. @fcb.param MI,Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates the map containing applicant data with applicant serial number as the key. @return FactInformationDTO[], Returns a list of facts required for execution of Low Doc Rule. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method needs to be called if LRM is applicable at the product group level and will do the following: <li>1.calculate LVR, execute rule to get submission type and provider type</li> <li>2.Execute rule to get if DUA is applicable</li> @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001   This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001   This exception is thrown when we try to execute LRM rules   for a Product Group for which LMI is not applicable. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011   This exception is thrown when Rule Id is not linked in Origination Preference  for LRM Submission Type and LRM Provider Rules.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method will be invoked to evaluate LRMSubmissionType rule to get the Submission type for the LRM policy. This will be called in the pre-assessment stage. @param sessionContext @param submissionInputDTO @param productGroupSerialNo @return @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method needs to be called if LRM is applicable at the product group level and will do the following: <li>1.Execute rule to check if LRM is required</li><li>2. If LRM is required,calculate LVR, execute rule to get submission type and provider type</li>@fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,CreditFacilityDTO[] facilityDetailsDTOs , This contains the various collaterals linked to the Product Group. @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method will update the approved amount in the credit facility dto from lcm to be 0  and the product group linkage from origination to be 0 @param sessionContext @param submissionDecisionDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:- <ul style="list-style-type: none"><li>Approve</li><li>Conditionally Approve</li><li>Refer</li><li>Refer Downsell</li><li>Decline</li><li>Continue</li></ul> Depending on the outcome of the call to decision interface, further processing of submission will take place. <2.3>, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline task. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response object of this call that contains the details of credit decision of submission and enitites within submission @exception  {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#SUBMISSION_NOT_FOUND) : FC_OR_DE_0055&lt;br&gt; This exception is thrown if submission is not found for the given submission id.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND) : FC_OR_DE_0075&lt;br&gt; This exception is thrown if submission history record is not found for the submission id which is being edited or amended.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_INT_SENSITIVITY_RATE_NOT_MNT) : FC_OR_CD_0026&lt;br&gt; This exception is thrown if interest sensitivity rate is not maintained for the lending product group and currency combination applied for in the submission.&lt;br&gt; {@link</p> <p>OriginationErrorConstants#OR_VALUATION_TYPE_NULL) : FC_OR_CD_0029&lt;br&gt; This exception is thrown for a collateral which requires valuation and if the no valuation type is recommended by the credit assessment engine.&lt;br&gt; @throws FatalException</p>
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws	This method validates whether the LVR value has been changed from the last time value. If no change in LVR value then it return false otherwise returns true. @param sessionContext @param lendingPgLinkageDTO @param collaterals @param applicantDatas @param creditFacilityDTOs @return Boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This service is used to create new facilities or update existing facilities.  The following activities will be carried out in sequence.    a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the  facility. LCM internally checks if an existing borrowing entity exists for the parties passed and   returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity.   b) Update the borrowing entity identifier against the facility.   c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates   linkages to collaterals linked to the facility. Also, simulations that are attached to the facility  either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO,submissionInputDTO,, Indicates the submission details containing the applicant and facility information. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.   @throws FatalException @exception   {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076   This error occurs when:  The borrowing entity identifier for the group of applicants applying for a facility could not be generated. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method will inquire LVR for a given product group. This method is being called in both post and pre assessment. @param submissionInputDTO @param lendingPgLinkageDTO @param creditFacilityDTOs @return lvr
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This service is used to fetch LVR value for the facility post assessment, on Structure solution when before an account is created or amended to set it as input for simulation which will affect pricing.  Internally, this service will call inquireLVRForProductGroup which is common for both fetch LVR pre and post assessment @param submissionId @param lendingPgLinkageDTO @param facilities @return lvr

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method will update the LRM and LVR value preassessment
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	calculateLVR	This method will call LCM to calculate Loan to Value Ratio using CreditFacilityDTO[]. This method will be called in the pre-assessment stage. @param sessionContext - contains information such as user id, branch, branch code, channel etc. @param primaryFacilityId @param creditFacilityDTOs, Contains the details of the collaterals linked to the product group, etc. @param collateralDTOs @param applicantDataDTOs @param fcGraphDTOs @return UniformLTVResponse @throws FatalException	This method will update the insurance amount for funding table whilst fetching funding table dto. In case of amend funding table , set the different in the amount. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	Populate the product details for Product groups which are not fulfilled, rejected or withdrawn. @param sessionContext @param productGroupLinkageDTO @param description @param applicationRequestType @param borrowingEntityId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	<p>This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched.</p> <p>@fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException</p>	<p>This method is used to fetch origination fees using the application id and facility id. The logic of the method is as follows</p> <ul style="list-style-type: none"> <li>Fetch pricing ticket number using the service</li> </ul> <p>@link{OriginationFeeApplicationService#fetchOriginationFees(SessionContext, HeaderDTO, LendingProductGroupLinkageDTO, ApplicantDataDTO, CreditFacilityDTO[], CollateralDTO[]).}</p> <p>The input of This service can be derived as follows</p> <ul style="list-style-type: none"> <li>SessionContext - SessionContext from the input HeaderDTO - Inquire application details using base application service @link{com.ofss.fc.app.origination.service.core.application.BaseApplicationService.inquireApplication} using the application id, input. Pass the header dto from the ApplicationDTO retrieved.</li> <li>LendingProductGroupLinkageDTO - Fetch product group linkage from the ApplicationDTO retrieved above, using the facility id, input. Set the PricingTicketNumberForBundleFees to the pg linkage dto, if IsNominatedForBundle() and PricingTicketNumberForBundleFees() is empty, by getting the new pricing ticket number by invoking the service</li> <li>@link{OriginationFeeApplicationService#createBundleSubscriptionPricingPlan}</li> <li>ApplicantDataDTO</li> <li>Fetch all applicants for the application using the helper method</li> <li>@link{LendingApplicationHelper#fetchAllApplicantsForApplicationId}</li> <li>Convert applicant DTO To applicant data DTO using the helper method</li> <li>@link{SubmissionInquiryHelper#populateApplicantDataDTO}</li> <li>CreditFacilityDTO[] - Fetch credit facility array DTO using the service</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> @link{LendingSimulationApplicationService#fetchFacilityHierarchy &lt;/li&gt; &lt;li&gt;CollateralDTO[] - Fetch collateral using the service @link{LendingApplicationService#fetchCollateralsForApplication}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Use the pricing ticket number and get the pricing details using the service @link{OriginationPricingAdapter#fetchFeesForTicketNumber} . The input facts can be populated using the methods @link{LendingApplicationService#populateFactsForOriginationFee} and @link{LendingApplicationService#populateFundingFactDefaultValues &lt;/li&gt; &lt;li&gt;Iterate WrappedPriceFeatureInquiryResponse #wrappedPriceFeatureDTO[], convert the WrappedPriceFeatureDTO to PricingFeeDetailDTO using the builder @link{PricingFeeDetailDTOBuilder#build &lt;/li&gt; &lt;li&gt;Set the pricing ticket number and PricingFeeDetailDTOs in the response&lt;/li&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the origination fees is to be retrieved @fcb.param MI,String,facilityId, This object holds the facility id, for which the origination fees is to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.pricing.PricingFeesResponse}, This contains the information on origination fees (pricingTicketNumber and pricingFeeDetails) @throws com.ofss.fc.infra.exception.FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will populate all facts such as Facility, Party etc. and make an inquiry from PMU to get all eligible offers. @param sessionContext @param mainPFLinkage @param applicationResponse @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This service will process and fetch the LRM quote post assessment where submission input dto is not present. Lending product group dto of the product group needed to fetch LRM also being sent to identify it with the new funding table details.This method is being called from funding table directly everytime capitalization values are being changed post assessment. LRMQuoteInquiryResponse being returned will contain both LRM quote and updated LVR @param sessionContext @param submissionId @param lendingPgLinkageDTO @return @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType   LRM required is obtained from Origination Preference. Lending Product Group Linkage related facts are   sent to the rule engine to execute rule for that rule Id whose response will be LRM Required or not. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType  LRM Submission is obtained from Origination Preference.LVR value and Lending  Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMSubmissionType
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType  LRM Submission and LRM Provider is obtained from Origination Preference.LVR value and Lending  Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   The response is populated in LRM rule Execution Response.  @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMRuleExecutionResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method calculates the automatic Debt Servicing Obligation (DSO) amounts for each facility decision to be used at UI as part of repayment summary. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, submissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will update the approved amount in the credit facility dto from lcm to be 0  and the product group linkage from origination to be 0 @param sessionContext @param submissionDecisionDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, submissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will fetch the facility hierarchy from LCM in case proposed facility exists. Else it'll return the facility hierarchy from draft.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	Method to return CreditFacilityDTO[] for a given application and productGroupLinkage. @param sessionContext @param applicationDTO @param productGroupLinkageDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will update the LRM and LVR value post assessment
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityHierarchy	<p>This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched.</p> <p>@fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException</p>	<p>This method will update the insurance amount for funding table whilst fetching funding table dto. In case of amend funding table , set the different in the amount. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteFacilityHierarchy	This service will delete the entire facility hierarchy. Origination creates facility and its linkages in the proposed state. If a submission is withdrawn or the submission is modified which results in deletion of a facility (user deletes the product group that contains the facility), this service is called to delete the hierarchy from the LCM system. @param sessionContext @param facilityId @throws FatalException	This service will be called during edit submission and check the following:  1> Fetch all lending product groups under a lending application.  2> Check if account opening has been done for all accounts within the product group. If done, mark the application for status update as PROCESSED.  3> If product group has not been rejected or withdrawn, check if product group linkage is present after the user has modified the submission.This is checked by comparing the old facility list and incoming new facility list.  4> If a facility that previously existed is not found in the incoming list, then delete the facility hierarchy from LCM. If LMI policy and/or auto refinance policies have been initiated for such a facility, cancel the policies.  5> Finally for applications containing processed as well as not processed product groups, delete the product group linkages that are not processed from the applications as these will be referred in the new applications generated in edit. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingApplication lendingApplication, , Indicates the lending application being checked for marking as PROCESSED. @fcb.param MI,List<String> newFacilityList, , Indicates the incoming new facility list. @return Boolean, Whether the application is to be marked as PROCESSED. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteFacilityHierarchy	This service will delete the entire facility hierarchy. Origination creates facility and its linkages in the proposed state. If a submission is withdrawn or the submission is modified which results in deletion of a facility (user deletes the product group that contains the facility), this service is called to delete the hierarchy from the LCM system. @param sessionContext @param facilityId @throws FatalException	This service will be called during edit submission and check the following:  1> Fetch all credit card product groups under a credit card application.  2> Check if account opening has been done for all accounts within the product group. If done, mark the application for status update as PROCESSED.  3> If product group has not been rejected or withdrawn, check if product group linkage is present after the user has modified the submission.This is checked by comparing the old facility list and incoming new facility list.  4> If a facility that previously existed is not found in the incoming list, then delete the facility hierarchy from LCM.   5> Finally for applications containing processed as well as not processed product groups, delete the product group linkages that are not processed from the applications as these will be referred in the new applications generated in edit. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardApplication creditCardApplication, , Indicates the lending application being checked for marking as PROCESSED. @fcb.param MI,List<String> newFacilityList, , Indicates the incoming new facility list. @return Boolean, Whether the application is to be marked as PROCESSED. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteFacilityHierarchy	This service will delete the entire facility hierarchy. Origination creates facility and its linkages in the proposed state. If a submission is withdrawn or the submission is modified which results in deletion of a facility (user deletes the product group that contains the facility), this service is called to delete the hierarchy from the LCM system. @param sessionContext @param facilityId @throws FatalException	This method is called from withdrawProductGroupLinkage in case the product group is of lending type. This service will delete the facility details that are linked with the lending Product group. It will cancel the LRM policy if initiated for this product group. It will also cancel the auto refinance policy if initiated for this product group. @param sessionContext @param lendingPgLinkage @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteFacilityHierarchy	This service will delete the entire facility hierarchy. Origination creates facility and its linkages in the proposed state. If a submission is withdrawn or the submission is modified which results in deletion of a facility (user deletes the product group that contains the facility), this service is called to delete the hierarchy from the LCM system. @param sessionContext @param facilityId @throws FatalException	This method is called from withdrawProductGroupLinkage in case the product group is of credit card type. This service will delete the facility details that are linked with the credit card Product group. It will delete the facility hierarchy. @param sessionContext @param lendingPgLinkage @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateFacilityApprovalDetail	This service will update the proposed facility with approval details like sanctioned amount and approved term at LCM once the submission has been approved either through decision engine or through manual credit decision screen. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApprovalDetailsDTO,approvalDetailsDTO, ,ApprovalDetailsDTO The ApprovalDetailsDTO consists of fields that requires approval for the facility. @throws FatalException	This service will update the proposed facility with approval details like sanctioned amount and approved term at LCM once the submission has been approved either through decision engine or through manual credit decision screen. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApprovalDetailsDTO,approvalDetailsDTO, ,ApprovalDetailsDTO The ApprovalDetailsDTO consists of fields that requires approval for the facility. @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		<p>This service initiates title search task for all collaterals linked to given application. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current title search status for the collateral is NOT_INITIATED</li> <li>2) Title search required is maintained for the collateral category in LCM Collateral Preferences</li> <li>3) There is no existing unexpired title search report present for the collateral in LCM</li> </ul> <p>@param SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012 This exception is thrown if submission decision is not found for the given submission id. @link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013 This exception is thrown if application decision is not found for the given application id. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateralCategoryPreference		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId,applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionDTO} ,collateralDecisionDtos, ,Collateral decision dtos for which the initiation is to be performed @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.InitiateCollateralTasksResponse} This object represents the response that is send with updated collateral decisions after valuation-initiation</p> <p>@exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateralCategoryPreference		<p>This service returns the list of collateral linked to given application or submission and for which valuation is to be initiated. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current valuation status for the collateral is NOT_INITIATED</li> <li>2) Valuation is required is returned by the decision engine for the collateral decision</li> <li>3) Valuation required is maintained for the collateral category in LCM Collateral Preferences</li> <li>4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM</li> </ul> <p>If application id is passed then collateral decisions linked to the given application only are checked otherwise all the collateral decisions for given submission are checked</p> <p>@param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc</p> <p>@param String,applicationId, applicationId Unique Identifier for the application</p> <p>@param String,submissionId, submissionId Unique Identifier for the submission</p> <p>@return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having</p> <ul style="list-style-type: none"> <li>list of collaterals for which valuation is to be initiated</li> </ul> <p>@exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012 This exception is thrown if submission decision is not found for the given submission id.</p> <p>{@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013 This exception is thrown if application decision is not found for the given application id.</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateralCategoryPreference		<p>This service returns the list of collateral linked to given application id or submission id and for which title search is to be initiated.&lt;br&gt; The following conditions are checked before initiating the task-&lt;br&gt; 1) Current title search status for the collateral is NOT_INITIATED&lt;br&gt; 2) Title search required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 3) There is no existing unexpired title search report present for the collateral in LCM&lt;br&gt; If application id is passed then collateral decisions linked to the given application only are checked&lt;br&gt; otherwise all the collateral decisions for given submission are checked &lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @param String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having &lt;br&gt; list of collaterals for which title search is to be initiated @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return boolean @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralCategoryPreference		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	convertProposedToConfirmed	This service will 1. Activate or confirm New Proposed Facilities 2. Collateral Group Linkage confirm 3. Confirm the linkage of Covenants, Conditions to the facility All the above can be achieved by the method convertProposedToCreditFacility(facilityId) of ProposedFacilityApplicationService @param sessionContext @param facilityId @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology.This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference.After that it will save the current state of simulation.Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders,add-on card holders, account preferences such as balance transfer details,delivery preferences,card membership details and also the preferences for each card holder such as emboss name,card background image @return {@link CreditCardAccountResponse} This class contains

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account id and account configuration DTO @throws FatalException
IOrigination LCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	convertProposedToConfirmed	This service will 1. Activate or confirm New Proposed Facilities 2. Collateral Group Linkage confirm 3. Confirm the linkage of Covenants, Conditions to the facility All the above can be achieved by the method convertProposedToCreditFacility(facilityId) of ProposedFacilityApplicationService @param sessionContext @param facilityId @throws FatalException	@param sessionContext @param applicationId @param prodGrpLinkage @param lcmAdapter @return @throws FatalException
IOrigination LCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	createOrSaveFacility	This service will update the information for the facilities passed. @param sessionContext @param creditFacilityDTO @throws FatalException	This service is used to create new facilities or update existing facilities.  The following activities will be carried out in sequence.    a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the  facility. LCM internally checks if an existing borrowing entity exists for the parties passed and   returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity.   b) Update the borrowing entity identifier against the facility.   c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates   linkages to collaterals linked to the facility. Also, simulations that are attached to the facility  either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>SubmissionInputDTO,submissionInputDTO , ,  Indicates the submission details containing the applicant and facility information. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. &lt;br&gt; @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076 &lt;br&gt; This error occurs when:&lt;br&gt; The borrowing entity identifier for the group of applicants applying for a facility could not be generated.&lt;br&gt;</p>
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	createOrSaveFacility	<p>This service will update the information for the facilities passed.  @param sessionContext @param creditFacilityDTO @throws FatalException</p>	<p>This service will update the details for Credit Facility details passed. It also updates the offer references&lt;br&gt; i.e. the details for offer under which simulations are done.&lt;br&gt;This service is used for: &lt;br&gt; 1. An existing active facility is being amended. In this case, the facility details are in proposed state and are being updated.&lt;br&gt; 2. A new facility is created in Proposed state for each lending product group. The submission is then later modified and details are edited. The limit, tenor or borrowing entity (parties applying for the limit can be added or removed) can be modified for this Proposed facility.&lt;br&gt; 3. New Sub facilities are being added to the main facility.&lt;br&gt; 4. New simulations added as a result of which offer references are to be updated. 5. Capitalization Details are Lending Product group level at updated. Note: This service is used by Lending SS and Credit Card SS to update Offer and facilities. Hence we are having both Offer Reference(common for credit card and lending PG) and Lending Product Group as input.  @fcb.param MI,SessionContext,sessionContext,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details are being updated. @fcb.param MI,OfferReferenceDTO[], offerReferenceDTOs, , indicates the list of offer references that are to be updated for the product group within an application. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_LINKAGE_NOT_FOUND} : FC_OR_DE_0053&lt;br&gt; This error occurs when:&lt;br&gt; For the application identifier and product group serial number, no product group details were not found.&lt;br&gt;</p>
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	createOrSaveFacility	This service will update the information for the facilities passed. @param sessionContext @param creditFacilityDTO @throws FatalException	This method will update the insurance amount for funding table whilst fetching funding table dto. In case of amend funding table , set the different in the amount. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateFacilityOnDecision	This service will update the information related to the decisions. A separate method is required so as to update only the decision related data. Currently the LTV was also calculated in normal decision updates. This service will cater only to decision related updates on the facility. @param sessionContext @param creditFacilityDTO @throws FatalException	This method will update the approved amount in the credit facility dto from lcm to be 0  and the product group linkage from origination to be 0 @param sessionContext @param submissionDecisionDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateFacilityOnDecision	This service will update the information related to the decisions. A separate method is required so as to update only the decision related data. Currently the LTV was also calculated in normal decision updates. This service will cater only to decision related updates on the facility. @param sessionContext @param creditFacilityDTO @throws FatalException	This service will update the information related to the decisions. A separate method is required so as to update only the decision related data. Currently the LTV was also calculated in normal decision updates. This service will cater only to decision related updates on the facility. @param sessionContext @param creditFacilityDTO @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateFacilityAmountAndLVRonDecision	This method is used by the Origination System to update the decision related data in the LCM system. The reason for a separate service is that it doesnt trigger a LTV calculation on this facility update. @param sessionContext @param facilityList	This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, , { @link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateFacilityAmountAndLVRonDecision	This method is used by the Origination System to update the decision related data in the LCM system. The reason for a separate service is that it doesn't trigger a LTV calculation on this facility update. @param sessionContext @param facilityList	This method is used by the Origination System to update the decision related data in the LCM system. The reason for a separate service is that it doesn't trigger a LTV calculation on this facility update. @param sessionContext @param facilityList
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	createOrFetchBorrowingEntity	This service will pass a list of party identifiers and fetch the borrowing entity. If the borrowing entity exists, the same is passed else a new borrowing entity is created and passed.  Borrowing entities are created and maintained in LCM system. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Map<Integer,ApplicantDataDTO>, applicantMap, , Indicates a map of applicants with applicant serial number as the key @fcb.param MI,ProductDetailsDTO, productDetailsDTO, , Indicates the product details. @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException	This service will pass a list of party identifiers and fetch the borrowing entity. If the borrowing entity exists, the same is passed else a new borrowing entity is created and passed.  Borrowing entities are created and maintained in LCM system. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Map<Integer,ApplicantDataDTO>, applicantMap, , Indicates a map of applicants with applicant serial number as the key @fcb.param MI,ProductDetailsDTO, productDetailsDTO, , Indicates the product details. @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	createOrFetchBorrowingEntity	<p>This service will pass a list of party identifiers and fetch the borrowing entity. If the borrowing entity exists, the same is passed else a new borrowing entity is created and passed.&lt;br&gt; Borrowing entities are created and maintained in LCM system. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Map&lt;Integer,ApplicantDataDTO&gt;, applicantMap, , Indicates a map of applicants with applicant serial number as the key @fcb.param MI,ProductDetailsDTO, productDetailsDTO, , Indicates the product details. @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException</p>	<p>This service is used to create new facilities or update existing facilities.&lt;br&gt; The following activities will be carried out in sequence.&lt;br&gt; &lt;br&gt; a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the&lt;br&gt; facility. LCM internally checks if an existing borrowing entity exists for the parties passed and &lt;br&gt; returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity. &lt;br&gt; b) Update the borrowing entity identifier against the facility. &lt;br&gt; c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates &lt;br&gt; linkages to collaterals linked to the facility. Also, simulations that are attached to the facility&lt;br&gt; either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO,submissionInputDTO , , Indicates the submission details containing the applicant and facility information. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. &lt;br&gt; @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076 &lt;br&gt; This error occurs when:&lt;br&gt; The borrowing entity identifier for the group of applicants applying for a facility could not be generated.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service returns a distinct list of collaterals for list of facility Ids passed @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionDTO submissionDTO, , Existing Submission Details @return CollateralDTO[], List of collaterals belonging to the facility list.
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method will get the collterals for the application @param Application application, the application domain object @return CollateralDTO list - collaterals for that application @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service fetches the Credit Decision for a collateral @param submissionId @param applicationId @param collateralId @return
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service fetches the Credit Decision for a collateral @param submissionId @param collateralId @return CollateralDecision
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service fetches the collateral decisions for a submission or application @param submissionId @param applicationId @return List<CollateralDecision>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method is used to fetch all collateral valuation summary for the application. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Fetch application decision using the service @link{CreditDecisionApplicationService#inquireApplicationDecision &lt;/li&gt; &lt;li&gt;Iterate collateral decisions from application decision retrieved above and collect all collateral ids&lt;/li&gt; &lt;li&gt;Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCollateralTitleSearchSummary and get the title search summary &lt;/li&gt; &lt;li&gt;Return the above response &lt;/li&gt; &lt;/ul&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Application Collateral Valuation Summary needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method is used to fetch all collateral title search summary for the application. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Fetch application decision using the service @link{CreditDecisionApplicationService#inquireApplicationDecision &lt;/li&gt; &lt;li&gt;Iterate collateral decisions from application decision retrieved above and collect all collateral ids&lt;/li&gt; &lt;li&gt;Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCollateralValuationSummary and get the valuation summary&lt;/li&gt; &lt;li&gt;Return the above response&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the Application Collateral Title Search Summary needs to be retrieved @return @link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralTitleSearchSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method is used to fetch origination fees using the application id and facility id. The logic of the method is as follows</p> <ul style="list-style-type: none"> <li>Fetch pricing ticket number using the service <code>@link{OriginationFeeApplicationService#fetchOriginationFees(SessionContext, HeaderDTO, LendingProductGroupLinkageDTO, ApplicantDataDTO, CreditFacilityDTO[], CollateralDTO[])}</code>. The input of This service can be derived as follows</li> <li><code>SessionContext</code> - <code>SessionContext</code> from the input <code>HeaderDTO</code> - Inquire application details using base application service <code>@link{com.ofss.fc.app.origination.service.core.application.BaseApplicationService.inquireApplication}</code> using the application id, input. Pass the header dto from the <code>ApplicationDTO</code> retrieved.</li> <li><code>LendingProductGroupLinkageDTO</code> - Fetch product group linkage from the <code>ApplicationDTO</code> retrieved above, using the facility id, input. Set the <code>PricingTicketNumberForBundleFees</code> to the pg linkage dto, if <code>IsNominatedForBundle()</code> and <code>PricingTicketNumberForBundleFees()</code> is empty, by getting the new pricing ticket number by invoking the service <code>@link{OriginationFeeApplicationService#createBundleSubscriptionPricingPlan}</code></li> <li>Fetch all applicants for the application using the helper method <code>@link{LendingApplicationHelper#fetchAllApplicantsForApplicationId}</code></li> <li>Convert applicant DTO To applicant data DTO using the helper method <code>@link{SubmissionInquiryHelper#populateApplicantDataDTO}</code></li> <li><code>CreditFacilityDTO[]</code> - Fetch credit facility array DTO using the service</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> @link{LendingSimulationApplicationService#fetchFacilityHierarchy &lt;/li&gt; &lt;li&gt;CollateralDTO[] - Fetch collateral using the service @link{LendingApplicationService#fetchCollateralsForApplication}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Use the pricing ticket number and get the pricing details using the service @link{OriginationPricingAdapter#fetchFeesForTicketNumber} . The input facts can be populated using the methods @link{LendingApplicationService#populateFactsForOriginationFee} and @link{LendingApplicationService#populateFundingFactDefaultValues &lt;/li&gt; &lt;li&gt;Iterate WrappedPriceFeatureInquiryResponse #wrappedPriceFeatureDTO[], convert the WrappedPriceFeatureDTO to PricingFeeDetailDTO using the builder @link{PricingFeeDetailDTOBuilder#build &lt;/li&gt; &lt;li&gt;Set the pricing ticket number and PricingFeeDetailDTOs in the response&lt;/li&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the origination fees is to be retrieved @fcb.param MI,String,facilityId, This object holds the facility id, for which the origination fees is to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.pricing.PricingFeesResponse}, This contains the information on origination fees (pricingTicketNumber and pricingFeeDetails) @throws com.ofss.fc.infra.exception.FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service returns a distinct list of collaterals for list of facility Ids passed @param sessionContext @param submissionDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method will populate the Submission Input DTO from an existing submission. For cases where the submission is modified, the user is taken back to the home page and is allowed to edit any information (whatever is permissible) previously entered. For product groups that are fulfilled i.e. the accounts have been opened, the corresponding applications are not allowed for any modification. For all other applications, the user edits the information required and re submits the submission. The Submission process is called and the input is passed again and corresponding modifications are done to the existing submission in the system.&lt;br&gt;This service is to be used only for lending product groups to be edited and should not be used to view summary of the entire submission.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,submissionId, ,The id of the submission for which details are being edited. @return An instance of {@link com.ofss.fc.app.origination.dto.core.submission.DraftSubmissionInquiryResponse} Contains the Submission Input details for submission Id passed. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method creates an array of {@link CollateralDTO} which contains information about the collateral which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link CollateralDTO} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method performs the post approval tasks including calculating approval expiry date, updating lending product group linkage and updating approval details for facility at LCM. @return @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		<p>This service initiates title search task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current title search status for the collateral is NOT_INITIATED&lt;br&gt; 2) Title search required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 3) There is no existing unexpired title search report present for the collateral in LCM&lt;br&gt; @param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionDTO} ,collateralDecisionDtos, Collateral decision dtos for which the initiation is to be performed @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.InitiateCollateralTasksResponse} This object represents the response that is send with updated collateral decisions after valuation-initiation</p> <p>@exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This service is called by the collateral valuation task to update the state of the collateral valuation Task in Origination. Each CollateralDecision entity stores the ValuationStateType and TitleSearchStateType. If the ValuationStateType is COMPLETED Or WAIVED and TitleSearchStateType is COMPLETED, then this service will update the EvaluationStateType that is also stored at CollateralDecision. Each ApplicationDecision entity stores the EvaluationStateType that is updated by this service when all the CollateralDecision within ApplicationDecision has EvaluationStateType as COMPLETED. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @param String,applicationId, ,applicationId Unique Identifier for the application @param ValuationStateType,valuationStateType, ,valuationStateType This attribute indicates the state of valuation task of the collateral @param CollateralApprovalDataDTO, , collateralApprovalDataDTO, , collateralApprovalDataDTO This attribute indicates the collateral level approval data DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				found for the given submission id.  {@link OriginationErrorConstants#OR_VALUATION_STATE_INCORRECT} : FC_OR_CD_0021  This exception is thrown if valuation state type in the call to this service does not equal COMPLETED  @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This methods logs and process alert/event related to collateral valuation activity.It will call publishOriginationActivity from OriginationEventApplicationService which will internally call registerActivity from AlertsNotificationHandler for logging of events.This method will be called after a collateral valuation process gets completed for given collateral. @param sessionContext @param submissionId @param collateralId
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service is called by the collateral title search task to update the state of the collateral title search Task in Origination. Each CollateralDecision entity stores the ValuationStateType and TitleSearchStateType. If the ValuationStateType is COMPLETED Or WAIVED and TitleSearchStateType is COMPLETED, then this service will update the EvaluationStateType that is also stored at CollateralDecision. Each ApplicationDecision entity stores the EvaluationStateType that is updated by this service when all the CollateralDecision within ApplicationDecision has EvaluationStateType as COMPLETED. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>String,applicationId, ,applicationId Unique Identifier for the application @param TitleSearchStateType,titleSearchStateType, ,titleSearchStateType This attribute indicates the state of title search task of the collateral @param CollateralApprovalDataDTO, collateralApprovalDataDTO, , collateralApprovalDataDTO This attribute indicates the collateral level approval data DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_WAIVE_TITLE_SEARCH_APPROVAL_REQUIRED} : FC_OR_CD_0024&lt;br&gt; This error is raised to send the collateral title search waiver for DCA approval based on rule configuration&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This service fetches the CollateralDecision for the given submissionId, applicationId and collateralId. If application and collateral id passed are null then all collaterals linked with the submission will be fetched. If collateral id passed is null then all collateral linked with submission and application will be fetched.</p> <p>@param SessionContext,sessionContext, SessionContext which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @param String,applicationId, ,applicationId Unique Identifier for the application @param String,collateralId, ,collateralId Unique Identifier for the collateral @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionResponse} The response object of this call that contains the collateral decision DTO @exception&lt;br&gt; {@link OriginationErrorConstants#OR_COLLATERAL_DECISION_NOT_FOUND} : FC_OR_CD_0025&lt;br&gt; This exception is thrown if collateral decision is not found for the given inputs.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This service returns the list of collateral linked to given application or submission and for which valuation is to be initiated. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current valuation status for the collateral is NOT_INITIATED</li> <li>2) Valuation is required is returned by the decision engine for the collateral decision</li> <li>3) Valuation required is maintained for the collateral category in LCM Collateral Preferences</li> <li>4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM</li> </ul> <p>If application id is passed then collateral decisions linked to the given application only are checked otherwise all the collateral decisions for given submission are checked</p> <p>@param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @param String,submissionId, submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having</p> <ul style="list-style-type: none"> <li>list of collaterals for which valuation is to be initiated</li> </ul> <p>@exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012 This exception is thrown if submission decision is not found for the given submission id.</p> <p>{@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013 This exception is thrown if application decision is not found for the given application id.</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		<p>This service returns the list of collateral linked to given application id or submission id and for which title search is to be initiated.&lt;br&gt; The following conditions are checked before initiating the task-&lt;br&gt; 1) Current title search status for the collateral is NOT_INITIATED&lt;br&gt; 2) Title search required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 3) There is no existing unexpired title search report present for the collateral in LCM&lt;br&gt; If application id is passed then collateral decisions linked to the given application only are checked&lt;br&gt; otherwise all the collateral decisions for given submission are checked &lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @param String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having &lt;br&gt; list of collaterals for which title search is to be initiated @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method creates an array of {@link CollateralDTO} which contains information about the collateral which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link CollateralDTO} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method will 1. Fetch the collaterals for each facility id in the input FacilityID Arraylist 2. Collate all the collaterals thus obtained, into a single list eliminating all duplicates @param SessionContext - contains information such as user id, branch, branch code, channel etc. ArrayList - A list of facilityId's @return ArrayList - List of all collaterals for all the facilityId's in the input
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method gets called when the withdraw Submission is selected. This service will do the following operations</p> <ul style="list-style-type: none"> <li>1. Validates for whether any account is opened under this submission. If no account gets opened, it can proceed further, otherwise throw proper error message</li> <li>2. Change the status of all the Applications under this submission to Withdrawn</li> <li>3. Close all the product groups under each application.</li> <li>4. If the product group is of 3. Delete all the facilities linked part of this submission</li> <li>4. Kill all the process which are initiated by this submission</li> <li>5. Finally change the status of Submission to Withdrawn</li> </ul> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, WithdrawSubmissionDetailsDTO, withdrawSubmissionDetailsDTO, , Indicates the instance of WithdrawSubmissionDetailsDTO.  @fcb.param MI, String, submissionId, , Indicates the identifier for the submission. @fcb.param MI,Boolean isSubmissionFeeNegotiated,,It is true when withdraw fees is negotiated or waived and further requires approval for WithdrawSubmission Task. @fcb.param MI, SubmissionApprovalDataDTO submissionApprovalDataDTO, Indicates the submission level Approval Data DTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @throws FatalException @exception {<a href="#">@link</a> OriginationErrorConstants#SUBMISSION_NOT_FOUNDED} : FC_OR_DE_0055  This error occurs</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_WITHDRAWN_NOT_ALLOWED} : FC_OR_WA_0001</p> <p>&lt;br&gt; This error occurs when:&lt;br&gt; The application under the submission is already withdrawn or processed. Such submissions cannot be withdrawn if any of the applications under it are already processed.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_ALREADY_WITHDRAWN} : FC_OR_WA_0002 &lt;br&gt; This error occurs when:&lt;br&gt; A submission is already withdrawn and the user tries to withdraw same submission again. This will happen when the same submission is being accessed through human task and application tracker.&lt;br&gt; The user inquires the submission through application tracker and then withdraws the submission from the human task and later returns to the tracker to withdraw.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_WITHDRAWN_NOT_ALLOWED} : FC_OR_WA_0004</p> <p>&lt;br&gt; This error occurs when:&lt;br&gt; Accounts have been opened under an application and the user tries to withdraw such a submission. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_CANNOT_WITHDRAWN} : FC_OR_WA_0005 &lt;br&gt; This error occurs when:&lt;br&gt; Accounts have been opened under an application and the user tries to withdraw such a submission. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateral		<p>This method gets called when the withdraw Submission is selected. This service will do the following operations</p> <ul style="list-style-type: none"> <li>1. Change the status of all the Applications under this submission to Withdrawn</li> <li>2. If the product group is of</li> <li>3. Delete all the facilities linked part of this submission</li> <li>3. Kill all the process which are initiated by this submission</li> <li>4. Finally change the status of Submission to Withdrawn</li> </ul> <p><b>@fcb.param</b> MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><b>@fcb.param</b> MI, WithdrawSubmissionDetailsDTO, withdrawSubmissionDetailsDTO, , Indicates the instance of WithdrawSubmissionDetailsDTO.</p> <p><b>@fcb.param</b> MI, String, submissionId, , Indicates the identifier for the submission.</p> <p><b>@fcb.param</b> MI,Boolean isSubmissionFeeNegotiated,,It is true when withdraw fees is negotiated or waived and further requires approval for WithdrawSubmission Task.</p> <p><b>@return</b> {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.</p> <p><b>@throws</b> FatalException @exception <a href="#">{@link OriginationErrorConstants#SUBMISSION_NOT_FOUND}</a> : FC_OR_DE_0055 This error occurs when: The details for the submission identifier were not found or the submission identifier is incorrect.</p> <p><a href="#">{@link OriginationErrorConstants#OR_APPLICATION_WITHDRAWN_NOT_ALLOWED}</a> : FC_OR_WA_0001 This error occurs when: The application under the submission is already withdrawn or</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>processed. Such submissions cannot be withdrawn if any of the applications under it are already processed.</p> <p><a href="#">{@link OriginationErrorConstants#SUBMISSION_ALREADY_WITHDRAWN}</a> : FC_OR_WA_0002  This error occurs when:  A submission is already withdrawn and the user tries to withdraw same submission again. This will happen when the same submission is being accessed through human task and application tracker.  The user inquires the submission through application tracker and then withdraws the submission from the human task and later returns to the tracker to withdraw.</p> <p><a href="#">{@link OriginationErrorConstants#PRODUCT_GROUP_WITHDRAWN_NOT_ALLOWED}</a> : FC_OR_WA_0004  This error occurs when:  Accounts have been opened under an application and the user tries to withdraw such a submission.</p> <p><a href="#">{@link OriginationErrorConstants#SUBMISSION_CANNOT_WITHDRAWN}</a> : FC_OR_WA_0005  This error occurs when:  Accounts have been opened under an application and the user tries to withdraw such a submission.</p>
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method will fetch all the collaterals that has been added for the applications under submission. @param sessionContext @param submissionId @exception  {@link OriginationErrorConstants#NO_COLLATERALS_FOUND_FOR_SUBMISSION} : FC_OR_CL_0002  This exception is raised if collateral decision is not found for the submission. 
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will fetch the collateral summary for the given submission id and collateral id @param sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param submissionID the id of the submission. @param collateralID the id of the collateral whose summary is to be fetched. @return the {@link CollateralLinkageSummaryResponse} containing the collateral linkage summary. @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will check and return if collateral perfection is required for any collateral in the submission @param sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param submissionId the id of the submission. @return the {@link CollateralPerfectionResponse} @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return boolean @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		Method to fetch CollateralDTO[] for the given application and set of collateral ids. @param sessionContext @param applicationDTO @param collateralsPerApplication @return
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method will fetch list of Stages and their corresponding status for the lending Application based on the seed data.   If status returned is UNKNOWN then specific handling is done to check if the stage is IN_PROGRESS, COMPLETED or NOT_APPLICABLE. @param sessionContext @param applicationDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		Fetches isTitleSearchRequired flag from Collateral preference @param sessionContext @param collateralId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will use LCM Adapter to call the service from LCM module to fetch the valuation details summary @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String ,collateralId, , This is the collateral id @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} , This class contains the titleSearch summary dtos @throws com.ofss.fc.infra.exception.FatalException @exception  {@link OriginationErrorConstants#OR_INV_INPUT} : FC_OR_DE_0019  This error occurs when the partyIds is null or empty
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This service will use LCM Adapter to call the service from LCM module to fetch the valuation details summary @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String ,collateralId, , This is the collateral id @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} , This class contains the valuation summary dtos @throws com.ofss.fc.infra.exception.FatalException @exception {@link OriginationErrorConstants#OR_INV_INPUT} : FC_OR_DE_0019  This error occurs when the partyIds is null or empty

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		This method is used to populate the Collateral DTO array in the post assessment stage. During the post assessment stage the collateral DTO will be passed as null for refreshPricingFees service. in order to populate the collateral facts we have to find out the main facility from the credit facility DTOs and from the main facility we have to take the credit facility collateral DTOs and fetch the respective collateral DTOs using the collateral code stored inside. @param sessionContext @param creditFacilityDTOs @return CollateralDTO[] @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	generateFacilityId	<p>This service will retrieve the next available facility identifier from LCM. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO, creditFacilityDTO, , Contains the credit facility fields such as requested amount and term for facility and purpose. @return String, facilityId, , Indicates the facility identifier generated. @throws FatalException</p>	<p>This service will generate the facility Id and default facility category fields. In Origination, only basic fields such as requested amount and term for facility, purpose&lt;br&gt; and variant information such as employee benefits required, whether facility is low doc&lt;br&gt; and capital subvention scheme is being used for the facility.&lt;br&gt; &lt;br&gt; Facility Category for the product group under which a facility is being setup is inquired&lt;br&gt; from PMU. Facility Category fields are further fetched using this facility category code&lt;br&gt; and fields are defaulted to the facility being setup.&lt;br&gt; &lt;br&gt; The next available facility identifier is generated in LCM system and fetched. The &lt;br&gt; facility will be created using this identifier. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ProductGroupLinkageDTO, productGroupLinkageDTO, , This contains the product group details for which facility has been requested @return CreditFacilityResponse, Returns the credit facility dto populated with default category fields and facility identifier @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_CANNOT_BE_SETUP} : FC_OR_DE_0077 &lt;br&gt; This error occurs when:&lt;br&gt; Facility is being setup for product groups that are not of the type - Assets.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_ID_NOT_GENERATED} : FC_OR_DE_0078 &lt;br&gt; This error occurs</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				when:  Facility Identifier generation fails or no identifier is returned by LCM service.    {@link OriginErrorConstants#FACILITY_CATEGORY_INVALID} : FC_OR_DE_0083   This error occurs when:  Facility Category Identifier is incorrect or no details are returned from LCM service. 
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateFacilityId	This service will retrieve the next available facility identifier from LCM. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO, creditFacilityDTO, , Contains the credit facility fields such as requested amount and term for facility and purpose. @return String, facilityId, , Indicates the facility identifier generated. @throws FatalException	This method would generate the new facility-id for the given {@link CreditFacilityDTO} @param creditFacilityDTO, instance of {@link CreditFacilityDTO} @return Facility Id @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	generateFacilityId	This service will retrieve the next available facility identifier from LCM. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO, creditFacilityDTO, , Contains the credit facility fields such as requested amount and term for facility and purpose. @return String, facilityId, , Indicates the facility identifier generated. @throws FatalException	This method would generate the new facility-id for the given {@link CreditFacilityDTO} @param creditFacilityDTO, instance of {@link CreditFacilityDTO} @return Facility Id @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	defaultFacilityCategoryFields	This service is used for defaulting facility category related fields in credit facility.  In Origination, only basic fields such as requested amount and term for facility, purpose  and variant information such as employee benefits required, whether facility is low doc  and capital subvention scheme is being used for the facility.    Facility Category for the product group under which a facility is being setup is inquired  from PMU. Facility Category fields are further fetched using this facility category code  and fields are defaulted to the facility being setup. @fcb.param MI,SessionContext,sessionContext,	This service will generate the facility Id and default facility category fields. In Origination, only basic fields such as requested amount and term for facility, purpose  and variant information such as employee benefits required, whether facility is low doc  and capital subvention scheme is being used for the facility.    Facility Category for the product group under which a facility is being setup is inquired  from PMU. Facility Category fields are further fetched using this facility category code  and fields are defaulted to the facility being setup.    The next available facility identifier is generated in LCM system and fetched. The   facility will be created using this identifier. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ProductGroupLinkageDTO, productGroupLinkageDTO, , This contains the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategory, , Indicates the facility category code @return CreditFacilityDTO, Returns the credit facility dto populated with default category fields @throws FatalException</p>	<p>product group details for which facility has been requested @return CreditFacilityResponse, Returns the credit facility dto populated with default category fields and facility identifier @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_CANNOT_BE_SETUP} : FC_OR_DE_0077 &lt;br&gt; This error occurs when:&lt;br&gt; Facility is being setup for product groups that are not of the type - Assets.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_ID_NOT_GENERATED} : FC_OR_DE_0078 &lt;br&gt; This error occurs when:&lt;br&gt; Facility Identifier generation fails or no identifier is returned by LCM service.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_CATEGORY_INVALID} : FC_OR_DE_0083 &lt;br&gt; This error occurs when:&lt;br&gt; Facility Category Identifier is incorrect or no details are returned from LCM service.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchRelationShipsForExistingParty	This method is used to fetch the related parties for the given parties. It will fetch the related parties from Party relationship table and also fetches the related parties from the borrowing entity table. This method is called the applicant details screen. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String[], partyIdList - Indicates the list of parties for which the relationship information are being fetched. @return {@link RelatedPartyInformationDTO} Array, Indicates the response obtained from party module	This method is used to fetch the related parties for the given parties and also the relationship between the parties. This method is called the applicant details screen. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String[],partyIdList, , List of party id's. The identifier which is used to uniquely identify a party with in the system. @return {@link FetchRelatedPartyInquiryResponse} An instance of the Fetch Related Party Inquiry Response which contains the relationship information that exist between two parties @param sessionContext @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchLTVHistory	This method is used to fetch the LTV history and is used for LTV deterioration check in post Create offerValidations. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Credit facility id @fcb.param MI, start date = credit decision created date @fcb.param, MI , end Date = current date @return FacilityLTVHistoryDetailsResponse	This method is used to fetch the LTV history and is used for LTV deterioration check in post Create offerValidations. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Credit facility id @fcb.param MI, start date = credit decision created date @fcb.param, MI , end Date = current date @return FacilityLTVHistoryDetailsResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchLTVHistory	This method is used to fetch the LTV history and is used for LTV deterioration check in post Create offerValidations. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Credit facility id @fcb.param MI, start date = credit decision created date @fcb.param, MI , end Date = current date @return FacilityLTVHistoryDetailsResponse	This method is used to fetch the LTV history and is used for LTV deterioration check in post Create offerValidations. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Credit facility id @fcb.param MI, start date = credit decision created date @fcb.param, MI , end Date = current date @return FacilityLTVHistoryDetailsResponse
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchLTVHistory	This method is used to fetch the LTV history and is used for LTV deterioration check in post Create offerValidations. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, Credit facility id @fcb.param MI, start date = credit decision created date @fcb.param, MI , end Date = current date @return FacilityLTVHistoryDetailsResponse	This method validates whether the LVR value has been changed from the last time value. If no change in LVR value then it return false otherwise returns true. @param sessionContext @param lendingPgLinkageDTO @param collaterals @param applicantDatas @param creditFacilityDTOs @return Boolean
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	executeConditionPolicyAndCreateLinkages		This service will call LCM to persist the execute the condition policy to obtain the DERIVED conditions.   The facts passed for executing the rules are filtered based on BE and party id, collateral id common across the applications. @param sessionContext @param submissionDecisionDTO @param submissionDTO @param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				submissionInputDTO @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeConditionPolicyAndCreateLinkages		This service will call LCM service to save the MANUAL condition linkages entered by user from application form as well as condition linkages returned by decision engine.   Before calling LCM the manual conditions are filtered to remove duplicate combinations for BE + Entity Id + Condition Code across the applications.   Origination will save the condition linkage id in ApplicationDecisionConditionLinkage. @param sessionContext @param submissionDecisionDTO @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeConditionPolicyAndCreateLinkages		This service will execute policies maintained in the product for verification, condition and covenants. It will also create the linkages for conditions added during application form processing. This service will be called from process after auto credit assessment is done. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param SubmissionDecisionDTO, submissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception  {@link OriginationErrorConstants#OR_DECISION_STAGE_INCORRECT} : FC_OR_CD_0017  This exception is thrown if submission decision stage is other than AUTO.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeConditionPolicyAndCreateLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateConditionLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateConditionLinkages		This service will call LCM service to save the MANUAL condition linkages entered by user from application form as well as condition linkages returned by decision engine.   Before calling LCM the manual conditions are filtered to remove duplicate combinations for BE + Entity Id + Condition Code across the applications.   Origination will save the condition linkage id in ApplicationDecisionConditionLinkage. @param sessionContext @param submissionDecisionDTO @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateConditionLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeCovenantPolicyAndCreateLinkages		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeCovenantPolicyAndCreateLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeCovenantPolicyAndCreateLinkages		<p>This service will execute policies maintained in the product for verification, condition and covenants. It will also create the linkages for conditions added during application form processing. This service will be called from process after auto credit assessment is done. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param SubmissionDecisionDTO, , { @link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @return { @link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.</p> <p>@exception&lt;br&gt; { @link OriginationErrorConstants#OR_DECISION_STAGE_INCORRECT} : FC_OR_CD_0017&lt;br&gt; This exception is thrown if submission decision stage is other than AUTO.&lt;br&gt; @throws FatalException</p>
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	executeCovenantPolicyAndCreateLinkages		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateCovenantLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateCovenantLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	populateCovenantLinkages		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateFacilityCategory	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	validateFacilityCategory	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException	This service performs null checks for defaulting product group fields to facility @param productGroupLinkageDTO @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	validateFacilityCategory	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException	This service performs null checks for defaulting product group fields to facility @param productGroupLinkageDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCreditExposureForApplicants	This service will fetch the credit exposure for the list of parties passed. @param sessionContext @param partyIds @param currency @return @throws FatalException	This service will populate the fact required for the Low Doc rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details @fcb.param MI, Map<String, CollateralDTO>, collateralMap, , Indicates the collateral details with collateral code as the key. Collaterals linked to the facility are fetched using this collateral code. @fcb.param MI,Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates the map containing applicant data with applicant serial number as the key. @return FactInformationDTO[], Returns a list of facts required for execution of Low Doc Rule. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCreditExposureForApplicants	This service will fetch the credit exposure for the list of parties passed. @param sessionContext @param partyIds @param currency @return @throws FatalException	This method executes rules to determine if DUA is applicable for LRM. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCreditExposureForApplicants	This service will fetch the credit exposure for the list of parties passed. @param sessionContext @param partyIds @param currency @return @throws FatalException	This service fetched total credit exposure for the applying parties @param sessionContext @param lendingStructureSolutionDTO @param lendingProductGroupLinkageDTO @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateralValuationTypeDescription	This service fetches the collateral Valuation Type description for a given code. @param sessionContext @param valuationTypeCode @return @throws FatalException	This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	populateProposedConditionLinkages	This service fetch & populates condition linkages for application, applicant, facility and collateral in applicationDecisionConditionLinkageDTO @param sessionContext @param applicationDecisionConditionLinkageDTOs @throws FatalException	This method will fetch details from LCM and populate proposed conditions and covenants with status as Proposed_New or Proposed_Amend in ConditionItemDetailsDTO array @fcb.param MI,SessionContext,sessionContext @fcb.param MI,SubmissionDecisionDTO,submissionDecisionDTO @returns MI, ConditionItemDetailsDTO[] @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	populateProposedCovenantLinkages	This service fetch & populates covenant lifor application, applicant, facility and collateral in applicationDecisionCovenantLinkageDTO @param sessionContext @param applicationDecisionCovenantLinkageDTOs @throws FatalException	This method will fetch details from LCM and populate proposed conditions and covenants with status as Proposed_New or Proposed_Amend in ConditionItemDetailsDTO array @fcb.param MI,SessionContext,sessionContext @fcb.param MI,SubmissionDecisionDTO,submissionDecisionDTO @returns MI, ConditionItemDetailsDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	isCollateralValuationValidReportExists		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	isCollateralValuationValidReportExists		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current valuation status for the collateral is NOT_INITIATED</li> <li>2) Valuation is required is returned by the decision engine for the collateral decision</li> <li>3) Valuation required is maintained for the collateral category in LCM Collateral Preferences</li> <li>4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM</li> </ul> <p>@param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012  This exception is thrown if submission decision is not found for the given submission id.  {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013  This exception is thrown if application decision is not found for the given application id.  @throws</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	isCollateralValuationValidReportExists		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current valuation status for the collateral is NOT_INITIATED</li> <li>2) Valuation is required is returned by the decision engine for the collateral decision</li> <li>3) Valuation required is maintained for the collateral category in LCM Collateral Preferences</li> <li>4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM</li> </ul> <p>@param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc</p> <p>@param {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionDTO} ,collateralDecisionDtos, ,Collateral decision dtos for thich the initiation is to be performed</p> <p>@return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.InitiateCollateralTasksResponse} This object represents the response that is send with updated collateral decisions after valuation-initiation</p> <p>@exception  {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012  This exception is thrown if submission decision is not found for the given submission id.  {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013  This exception is thrown if application decision is not found for the given application id.  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	isCollateralValuationValidReportExists		<p>This service returns the list of collateral linked to given application or submission and for which valuation is to be initiated. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current valuation status for the collateral is NOT_INITIATED</li> <li>2) Valuation is required is returned by the decision engine for the collateral decision</li> <li>3) Valuation required is maintained for the collateral category in LCM Collateral Preferences</li> <li>4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM</li> </ul> <p>If application id is passed then collateral decisions linked to the given application only are checked otherwise all the collateral decisions for given submission are checked</p> <p>@param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc</p> <p>@param String,applicationId, applicationId Unique Identifier for the application</p> <p>@param String,submissionId, submissionId Unique Identifier for the submission</p> <p>@return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having</p> <ul style="list-style-type: none"> <li>list of collaterals for which valuation is to be initiated</li> </ul> <p>@exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012 This exception is thrown if submission decision is not found for the given submission id.</p> <p>{@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013 This exception is thrown if application decision is not found for the given application id.</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	isCollateral TitleSearch ValidReportExists		<p>This service initiates title search task for all collaterals linked to given application. The following conditions are checked before initiating the task-</p> <ul style="list-style-type: none"> <li>1) Current title search status for the collateral is NOT_INITIATED</li> <li>2) Title search required is maintained for the collateral category in LCM Collateral Preferences</li> <li>3) There is no existing unexpired title search report present for the collateral in LCM</li> </ul> <p>@param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012 This exception is thrown if submission decision is not found for the given submission id. {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013 This exception is thrown if application decision is not found for the given application id. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	isCollateralTitleSearchValidReportExists		<p>This service returns the list of collateral linked to given application id or submission id and for which title search is to be initiated.&lt;br&gt; The following conditions are checked before initiating the task-&lt;br&gt; 1) Current title search status for the collateral is NOT_INITIATED&lt;br&gt; 2) Title search required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 3) There is no existing unexpired title search report present for the collateral in LCM&lt;br&gt; If application id is passed then collateral decisions linked to the given application only are checked&lt;br&gt; otherwise all the collateral decisions for given submission are checked &lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @param String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralTaskInitiationResponse} A response object having &lt;br&gt; list of collaterals for which title search is to be initiated @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchProposedFacilityInformation		This method will construct the Account details DTO . If settlement has already been done for the application, the settlement reference and the linkages stored in origination will be used to construct the DTO. @param sessionContext @param lendingApplication @param settlementRef @return AccountDetailsDTO[] @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchProposedFacilityInformation		This method will construct the fee details for each Product group in the application. The fees are obtained from the funding table and pricing @param dto @return FeeDetailsDTO
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchProposedFacilityInformation		This method is used to build FacilityApprovalDataDTO
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchProposedFacilityInformation		This service will convert product details to product group linkage dtos and fetch and set the borrowing entity id for the PG from LCM for lending products. @param sessionContext @param productGroupLinkageDTOList @param productDetailsDTO @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchConfirmedFacilityInformation		This method will construct the Account details DTO . If settlement has already been done for the application, the settlement reference and the linkages stored in origination will be used to construct the DTO. @param sessionContext @param lendingApplication @param settlementRef @return AccountDetailsDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchConfirmedFacilityInformation		This method will construct the fee details for each Product group in the application. The fees are obtained from the funding table and pricing @param dto @return FeeDetailsDTO
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchGuarantorsforFacility		This service will fetch the guarantors for a particular facility. It will check with LCM for the guarantee collaterals linked to this facility and then for all those owners fetch the party details from MDM @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return MDMPartiesInquiryResponse , Response containing details of all guarantors for the facility. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchBorrowingEntityName		This service will return the borrowing entity description for borrowing entity passed. @param borrowingEntity @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	inquireFacilityConditions		This method checks if the preliminary conditions for the borrowing entity have been complied @throws FatalException @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the id of the lending application for which the create offer stage is in progress @fcb.param MI,LendingProductGroupLinkageDTO prodGroupLinkage, Indicates product group linkage @fcb.param MI,List<ConditionsType> conditionTypes, Conditions having condition type from this list of condition types   are to be checked for compliance @return List<ValidationError>
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	inquireCollateralConditions		This method checks if the preliminary conditions for the borrowing entity have been complied @throws FatalException @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the id of the lending application for which the create offer stage is in progress @fcb.param MI,LendingProductGroupLinkageDTO prodGroupLinkage, Indicates product group linkage @fcb.param MI,List<ConditionsType> conditionTypes, Conditions having condition type from this list of condition types   are to be checked for compliance @return List<ValidationError>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	inquirePartyConditions		This method checks if the preliminary conditions for the borrowing entity have been complied @throws FatalException @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the id of the lending application for which the create offer stage is in progress @fcb.param MI,LendingProductGroupLinkageDTO prodGroupLinkage, Indicates product group linkage @fcb.param MI,List<ConditionsType> conditionTypes, Conditions having condition type from this list of condition types   are to be checked for compliance @return List<ValidationError>
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyConditionLinkage	This method deletes the ProposedPartyConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads all the covenants and conditions linked to the submission which entity type is either Collateral or Applicant  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @param sessionContext @param submissionId @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyConditionLinkage	This method deletes the ProposedPartyConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyConditionLinkage	This method deletes the ProposedPartyConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedPartyConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralConditionLinkage	This method deletes the ProposedCollateralConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads all the covenants and conditions linked to the submission which entity type is either Collateral or Applicant  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @param sessionContext @param submissionId @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralConditionLinkage	This method deletes the ProposedCollateralConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralConditionLinkage	This method deletes the ProposedCollateralConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedCollateralConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyCovenantLinkage	This method deletes the ProposedPartyCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads all the covenants and conditions linked to the submission which entity type is either Collateral or Applicant  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @param sessionContext @param submissionId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyCovenantLinkage	This method deletes the ProposedPartyCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedPartyCovenantLinkage	This method deletes the ProposedPartyCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedPartyCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralCovenantLinkage	This method deletes the ProposedCollateralCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads all the covenants and conditions linked to the submission which entity type is either Collateral or Applicant  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @param sessionContext @param submissionId @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralCovenantLinkage	This method deletes the ProposedCollateralCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedCollateralCovenantLinkage	This method deletes the ProposedCollateralCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedCollateralCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedFacilityConditionLinkage	This method deletes the ProposedFacilityConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedFacilityConditionLinkage	This method deletes the ProposedFacilityConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedFacilityConditionLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedFacilityCovenantLinkage	This method deletes the ProposedFacilityCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method performs the following sequence of operations.  1. Loads the covenants and conditions linked to the submission based on deleteManualFlg flag passed in input. If flag is true then all linkages will be fetched else only External and Derived linkages will be fetched.  2. Iterates over the list of covenants and conditions and checks whether the same linkage is linked to some other submission or not.  3. If the linkage is already linked to some other submission then do nothing.  4. If the linkage is not linked to any other submission then remove the linkage from Credit decision tables as well as from LCM tables.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission Id for which covenants and conditions are to be deleted. @fcb.param MI, Boolean deleteManualFlg, , Indicates if manually added covenants and conditions are to be deleted or not. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteProposedFacilityCovenantLinkage	This method deletes the ProposedFacilityCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException	This method deletes the ProposedFacilityCovenantLinkage for the given linkage id. @param sessionContext @param linkageId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This service will be called on change of borrowers scenario to check if any bundles are linked to the accounts which are getting affected. If new parties are getting linked to accounts which are already part of bundle then the parties should be added as bundle owners as well. This service will return the bundle details and an array of bundled accounts which are impacted per bundle.  The following steps are performed-  i) All bundles with bundle owners as parties in the old parties list are fetched  ii) If bundle status is Active or Inactive, check that at least one party is present in new parties list which was not present as existing Bundle owner  iii) Create list of credit accounts under the facility being amended which are part of the above bundles  iv) It will also create a list of bundle owners based on the union of new owners of accounts which are linked to bundle as well as existing bundle owners.   @param sessionContext @param facilityBundleDetailsRequest @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	<p>This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched.</p> <p>@fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method constructs a {@link SubmissionInputDTO} for the given Facility Id. The facility id should be the id of the parent facility. This method creates the Submission Header from the contextual information present in {@link SessionContext}. It also generates a new Submission Id for the Header.  It inquires PMU for information on the products linked to this facility and constructs a list of {@link ProductDetailsDTO} which forms a part of the {@link SubmissionInputDTO}.  It inquires MDM for information on the applicants linked to this facility and constructs a list of {@link ApplicantDataDTO} which forms a part of the {@link SubmissionInputDTO}.  It inquires LCM and fetches the list of collateral {@link CollateralDTO} linked to facility which forms a part of the {@link SubmissionInputDTO}.  Other attributes of {@link SubmissionInputDTO} are left to its default values. @param sessionContext the {@link SessionException} @param facilityId the facility Id of the main parent facility @param accountCode the Account Id of the account opened under the facility @return {@link SubmissionInputDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	Validates whether the Collateral facility linkage has been changed from last time. If there is no new collateral added or existing collateral is modified the system will return false, otherwise returns true @param sessionContext @param creditFacilityDTO @param facilityId @return Boolean
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	Validates whether the Collateral facility linkage has been changed from last time. If there is no new collateral added or existing collateral is modified the system will return false, otherwise returns true @param sessionContext @param creditFacilityDTO @param facilityId @return Boolean
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	Validates whether the Collateral facility linkage has been changed from last time. If there is no new collateral added or existing collateral is modified the system will return false, otherwise returns true @param sessionContext @param creditFacilityDTO @param facilityId @return Boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	This method will update the LRM and LVR value post assessment

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchActiveFacilityHierarchy	This service will fetch the facility hierarchy i.e. all facilities and sub facilities created under facility or sub facility details will be fetched. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return FacilityInquiryResponse , Response containing details of facility and all other sub facilities under the given facility. @throws FatalException	getting original requested amount from the main line effective limit amount @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityBalanceDetails	This method fetches the balance details of the facility id @param sessionContext @param facilityBalanceDetailsDTO @return @throws FatalException	
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityBalanceDetails	This method fetches the balance details of the facility id @param sessionContext @param facilityBalanceDetailsDTO @return @throws FatalException	
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	fetchFacilityBalanceDetails	This method fetches the balance details of the facility id @param sessionContext @param facilityBalanceDetailsDTO @return @throws FatalException	This method fetches the balance details of the facility id @param sessionContext @param facilityBalanceDetailsDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralValuationSummary	<p>This will <b>&lt;b&gt;return the summary details&lt;/b&gt;</b> which will be a combination of the fields in the main table</p> <p><b>&lt;b&gt;FLX_LM_CVAL_REQUEST_B&lt;/B&gt;</b> as well as the interface table <b>&lt;FLX_LM_CVAL_INTF_B&lt;/b&gt;</b>.&lt;br&gt; &gt; There will be as many number of rows in the array, as there are entries in the interface table.</p> <p>@param sessionContext @param collateralId @return @throws FatalException</p>	<p>This method is used to fetch all collateral valuation summary for the application. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Fetch application decision using the service @link{CreditDecisionApplicationService#inquireApplicationDecision &lt;/li&gt; &lt;li&gt;Iterate collateral decisions from application decision retrieved above and collect all collateral ids&lt;/li&gt; &lt;li&gt;Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCollateralTitleSearchSummary and get the title search summary &lt;/li&gt; &lt;li&gt;Return the above response &lt;/li&gt; &lt;/ul&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the Application Collateral Valuation Summary needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchCollateralValuationSummary	<p>This will <b>&lt;b&gt;return the summary details&lt;/b&gt;</b> which will be a combination of the fields in the main table</p> <p><b>&lt;b&gt;FLX_LM_CVAL_REQUEST_B&lt;/B&gt;</b> as well as the interface table <b>&lt;FLX_LM_CVAL_INTF_B&lt;/b&gt;</b>.&lt;br&gt; &gt; There will be as many number of rows in the array, as there are entries in the interface table.</p> <p>@param sessionContext @param collateralId @return @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchCollateralValuationSummary	This will <b>return the summary details</b> which will be a combination of the fields in the main table <b>FLX_LM_CVAL_REQUEST_B</B> as well as the interface table <FLX_LM_CVAL_INTF_B</b>. > There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException	This service will use LCM Adapter to call the service from LCM module to fetch the valuation details summary @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String ,collateralId, , This is the collateral id @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} , This class contains the valuation summary dtos @throws com.ofss.fc.infra.exception.FatalException @exception {@link OriginationErrorConstants#OR_INV_INPUT} : FC_OR_DE_0019  This error occurs when the partyIds is null or empty
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchTitleSearchSummary	This will return the <u>summary details</u> which will be a combination of the fields in the main table <b>flx_lm_ttls_request_b</b> as well as the interface table <b>flx_lm_ttls_intf_b</b>. There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchTitleSearchSummary	This will return the <u>summary details</u> which will be a combination of the fields in the main table <b>flx_lm_ttls_request_b</b> as well as the interface table <b>flx_lm_ttls_intf_b</b>. There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException	This will return the <u>summary details</u> which will be a combination of the fields in the main table <b>flx_lm_ttls_request_b</b> as well as the interface table <b>flx_lm_ttls_intf_b</b>. There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchTitleSearchSummary	This will return the <u>summary details</u> which will be a combination of the fields in the main table <b>flx_lm_ttls_request_b</b> as well as the interface table <b>flx_lm_ttls_intf_b</b>. There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchTitleSearchSummary	This will return the <u>summary details</u> which will be a combination of the fields in the main table <b>flx_lm_ttls_request_b</b> as well as the interface table <b>flx_lm_ttls_intf_b</b>. There will be as many number of rows in the array, as there are entries in the interface table. @param sessionContext @param collateralId @return @throws FatalException	This service will use LCM Adapter to call the service from LCM module to fetch the valuation details summary @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String ,collateralId, , This is the collateral id @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralValuationSummaryResponse} , This class contains the titleSearch summary dtos @throws com.ofss.fc.infra.exception.FatalException @exception  {@link OriginationErrorConstants#OR_INV_INPUT} : FC_OR_DE_0019  This error occurs when the partyIds is null or empty
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	cancelValuationRequest	This method <u>Cancels</u> the valuation request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException	This method <u>Cancels</u> the valuation request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	cancelValuationRequest	This method <u>Cancels</u> the valuation request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	cancelTitleSearchRequest	This method <u>Cancels</u> the TitleSearch request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException	This method <u>Cancels</u> the TitleSearch request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	cancelTitleSearchRequest	This method <u>Cancels</u> the TitleSearch request for the given request message reference . @param sessionContext @param referenceId @return @throws FatalException	
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	inquireCreditFacility		Retrieve number of days for account opening or first disbursement based on the product group / credit policy details @param sessionContext @param facilityId @param productGroupLinkageType @param creditPolicyDTO @param creditFacility @return accountOpeningOrFirstDisbursementDays @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	inquireCreditFacility		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	inquireCreditFacility		This method will validate if one account has been nominated for facility fee @param creditFacilityDTOs @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	inquireCreditFacility		This method is invoked post-assessment structure solution. It is validated for the facility expiry for AMEND applications. The facility availability date, if present, should be after the value-date. If availability date is not present then the facility-period is checked against the value date @param sessionContext @param valueDate @param lendingProductGroupLinkageDTO @return List of validation errors

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	inquireCreditFacility		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	inquireProposedFacility		This method performs the validations listed below: <li>Rejected product groups - if any of the product group related offer document is rejected, the product group is marked as 'Offer_REJECTED' and withdrawn.</li> <li>LTV code deterioration - If LTV has deteriorated for any product group , Collateral valuation and Collateral title search @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,Set&lt;OfferDocumentLinkage&gt; all the offer documents for the application @return List&lt;ValidationError&gt; if any</li>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	inquireProposedFacility		This service checks for collateral valuation completion, LVR and LTV code deterioration   based on the following cases:  If NCCP = yes and LRM condition code does not exist or is waived, valuation must be completed for all  collateral and LTV code deterioration check must be done.  If NCCP = yes and LRM condition code exists, valuation must be completed for all collateral, LVR   threshold breach must be checked and LTV code deterioration check must be done.  If NCCP = no and LRM condition code exists, valuation must be completed for all collateral and LVR   threshold breach must be checked.  If NCCP = no and LRM condition code does not exist or is waived, no check is required.  The LRM_CONDITION_CODE is configured in the OriginationConfiguration properties file.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, ,lendingProductGroupLinkageDTO, , Indicates the lending product group for which LTV deterioration is being verified. @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	saveConditionLinkageSummary	@param sessionContext @param conditionLinkageSummaryWrapperDTO @return @throws FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	saveCovenantLinkageSummary	@param sessionContext @param covenantLinkageWrapperDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateNominatedAccountForFacilityFees		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	updateNominatedAccountForFacilityFees		@param sessionContext @param prodGrpLinkage @param accountOpeningDetailsDTO @return @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	This API performs change of Owners and change nomination of account during amend flow in bundles. The Amend of Bundle occurs based on action passed in the {@link AmendBundleDTO} . The two values that can be passed are <ol style="list-style-type: none"> <li>{@link AmendBundleAction#CHANGE_OWNERS}</li> <li>{@link AmendBundleAction#DELINK_ACCOUNT}</li> </ol> It performs following functions <ol style="list-style-type: none"> <li>For the reference id passed the corresponding bundle is fetched</li> <li>If not found a new one is created</li> <li>a list of party Id is created from the current submission which is being amended</li> <li>If bundle action is set to amend the partyIds are set and ComponentPreferenceDTO is set to null</li> <li>else the account which needs to be delinked is fetched and the account details are set with the updated componentPreferenceDTO which is populated from the settlement Taskflow</li> </ol> Following are the Prerequisites for using this method... <ol style="list-style-type: none"> <li>{@link SubmissionCreationApplicationService#generateSubmissionId(SessionContext, com.ofss.fc.app.origination.dto.core.submission.Heade</li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> rDTO)} &lt;/li&gt; &lt;li&gt; {@link SubmissionCreationApplicationService#amendLendingFacility(SessionContext, String, String)&lt;/li&gt; &lt;/ol&gt; @param sessionContext MI,SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. Session Context must contain the session context information such as user id, branch, branch code, channel etc. @param submissionId MI,String,submissionId, contains the submission id against which the bundles are managed. @param amendBundleDTO contains the following attributes &lt;ol&gt; &lt;li&gt;bundleReference&lt;/li&gt; &lt;li&gt;amendBundleAction&lt;/li&gt; &lt;li&gt;accountDetails&lt;/li&gt; &lt;li&gt;nominatedAccountId&lt;/li&gt; &lt;li&gt;partyIds&lt;/li&gt; &lt;/ol&gt; @return Transaction Status.The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; @throws {@link com.ofss.fc.infra.exception.FatalException} Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	This method creates an array of {@link ApplicantDataDTO} which contains information about the applicants which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link ApplicantDataDTO} @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	This method creates an array of {@link ApplicantDataDTO} which contains information about the applicants which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link ApplicantDataDTO} @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	This method will fetch the party to party relationships between all parties for the given borrowing entity id @param sessionContext the {@link SessionContext} @param borrowingEntityId the borrowing entity id @return an array of {@link PartyToPartyRelationshipDTO} @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	@see IOriginationLCMAAdapter#fetchPartyForBorrowingEntity(SessionContext, String)
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchPartyForBorrowingEntity	This method is used to fetch the list of party id for the given borrowing entity id @param sessionContext @param borrowingEntityCode @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	saveFacilityChargeGraphs	This method is used to save the facility charge graph list for submission @param sessionContext @param facilityChargeGraphDTOs @return @throws FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	saveFacilityChargeGraphs	This method is used to save the facility charge graph list for submission @param sessionContext @param facilityChargeGraphDTOs @return @throws FatalException	This method is used to persist the Facility Charge Graphs along with updating decision details for the given proposed facility Ids which are part of the submission
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	saveFacilityChargeGraphs	This method is used to save the facility charge graph list for submission @param sessionContext @param facilityChargeGraphDTOs @return @throws FatalException	This service is used to create new facilities or update existing facilities.  The following activities will be carried out in sequence.    a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the  facility. LCM internally checks if an existing borrowing entity exists for the parties passed and   returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity.   b) Update the borrowing entity identifier against the facility.   c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates   linkages to collaterals linked to the facility. Also, simulations that are attached to the facility  either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO,submissionInputDTO,, Indicates the submission details containing the applicant and facility information. @return {@link com.ofss.fc.service.response.TransactionStatus} An

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.   @throws FatalException @exception   {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076   This error occurs when:  The borrowing entity identifier for the group of applicants applying for a facility could not be generated. 
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateCollateral		This method checks for the following Consumer Protection Act (NCCP) validations The following validations have to be successful 1. The verification for the application has to be completed successfully 2. The title search if required has to be completed successfully. @throws Exception @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , The lending application for which the create offer stage is in progress @fcb.param MI, LendingProductGroupLinkage[], the array of product group linkages for the application
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateCollateral		This method determines whether the title search has been done for all the collaterals submitted for the application @param collateralDecisions @return Boolean true if title search has been done for all the collaterals @throws FatalException @throws Exception
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateCollateral		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	validateCollateral		<p>This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:-</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Approve&lt;/li&gt;</li> <li>&lt;li&gt;Conditionally Approve&lt;/li&gt;</li> <li>&lt;li&gt;Refer&lt;/li&gt;</li> <li>&lt;li&gt;Refer Downsell&lt;/li&gt;</li> <li>&lt;li&gt;Decline&lt;/li&gt;</li> <li>&lt;li&gt;Continue&lt;/li&gt;</li> </ul> <p>Depending on the outcome of the call to decision interface, further processing of submission will take place. &lt;2.3&gt;, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline task. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@return {@link com.ofss.fc.app.Origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response object of this call that contains the details of credit decision of submission and entities within submission @exception&lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055&lt;br&gt; This exception is thrown if submission is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND} : FC_OR_DE_0075&lt;br&gt; This exception is thrown if submission history record is not found for the submission id which is being edited or amended.&lt;br&gt; {@link OriginationErrorConstants#OR_INT_SENSITIVITY_RATE_NOT_MNT} : FC_OR_CD_0026&lt;br&gt; This exception is thrown if interest sensitivity rate is not maintained for the lending product group and currency combination applied for in the submission.&lt;br&gt; {@link OriginationErrorConstants#OR_VALUATION_TYPE_NULL} : FC_OR_CD_0029&lt;br&gt; This exception is thrown for a collateral which requires valuation and if the no valuation type is recommended by the credit assessment engine.&lt;br&gt; @throws FatalException</p>
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	validateCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	validateCollateral		This service checks for collateral valuation completion, LVR and LTV code deterioration   based on the following cases:  If NCCP = yes and LRM condition code does not exist or is waived, valuation must be completed for all  collateral and LTV code deterioration check must be done.  If NCCP = yes and LRM condition code exists, valuation must be completed for all collateral, LVR   threshold breach must be checked and LTV code deterioration check must be done.  If NCCP = no and LRM condition code exists, valuation must be completed for all collateral and LVR   threshold breach must be checked.  If NCCP = no and LRM condition code does not exist or is waived, no check is required.  The LRM_CONDITION_CODE is configured in the OriginationConfiguration properties file.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group for which LTV deterioration is being verified. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	createFacilityFeeRecommendation	This service will create the facility fee recommendation and return the ticket number @param sessionContext @param creditFacilities @return @throws FatalException	This method will validate if one account has been nominated for facility fee @param creditFacilityDTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	addOrUpdateCollateral	This method adds or updates the given collateral @param sessionContext @param collateralDTO @return TransactionStatus @throws FatalException	
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	addOrUpdateCollateral	This method adds or updates the given collateral @param sessionContext @param collateralDTO @return TransactionStatus @throws FatalException	This service represents the addOrUpdateCollateral method of the Collateral Application Service. This method is used for create/update multiple new collateral's into the system by capturing the details like Collateral Type,Collateral Category,Collateral sub category,Owner Estimated Market Value,Market Value,Bank Value and Bank Value relied on etc., And this service will also captures the Address details and ownership details of the collateral and supports the all types(Property,Investments,Automobile,FixedAssest,Cash&Deposit and Guarantee etc.,) of collateral's to create into the system.  System will automatically generates unique id for every newly creating collateral.   While creating a collateral, system will retrieves the details from the collateral category preferences(like Bank haircut(%),Categorization Haircut(%),Charge Required) defined for the specified collateral type and collateral category and calculates the Bank Value,Category value,Bank Value Relied on and Capitalization Value.  And for newly creating collateral De-dupe search also performs to show the similar collateral's existing in the system to the end user.   @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO TODO: Explain the collateralDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination LCMAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nLCMAdapter	validateSan ctionedAm ountForHie rarchy		
IOrigination LCMAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nLCMAdapter	validateSan ctionedAm ountForHie rarchy		
IOrigination LCMAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nLCMAdapter	fetchCollate ralValuatio nReports		This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException
IOrigination LCMAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio	fetchDirect ExposureD etails		This service will return facts required for pricing during structure solution in pre assessment for all lending product groups in the submission. Currently total direct credit exposure is being populated. It is



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																								
	nLCMAdapter			<p>computed in LCM for existing parties and proposed and credit facilities. &lt;br&gt; The inputs will be sent from Origination based on below logic. Consider the following scenario. A is existing party and B is new party.&lt;br&gt;</p> <table border="1"> <tr> <td>&lt;b&gt;Party&lt;/b&gt;&lt;/td&gt;</td> <td>&lt;b&gt;Product Group&lt;/b&gt;&lt;/td&gt;</td> <td>&lt;b&gt;Requested Amount&lt;/b&gt;&lt;/td&gt;</td> </tr> <tr> <td>A&lt;/td&gt;</td> <td>Home Loan&lt;/td&gt;</td> <td>100k&lt;/td&gt;</td> </tr> <tr> <td>B&lt;/td&gt;</td> <td>Personal Loan&lt;/td&gt;</td> <td>50k&lt;/td&gt;</td> </tr> <tr> <td>A, B&lt;/td&gt;</td> <td>Over Draft&lt;/td&gt;</td> <td>150k&lt;/td&gt;</td> </tr> </table> <p>The following input will be passed to LCM in order to calculate the total direct credit exposure for each product group&lt;br&gt;</p> <table border="1"> <tr> <td>&lt;b&gt;For Product Group&lt;/b&gt;&lt;/td&gt;</td> <td>&lt;b&gt;Party&lt;/b&gt;&lt;/td&gt;</td> <td>&lt;b&gt;Facility&lt;/b&gt;&lt;/td&gt;</td> </tr> <tr> <td>Home Loan&lt;/td&gt;</td> <td>A&lt;/td&gt;</td> <td>Home Loan, Over Draft&lt;/td&gt;</td> </tr> <tr> <td>Personal Loan&lt;/td&gt;</td> <td>B&lt;/td&gt;</td> <td>Personal Loan, Over Draft&lt;/td&gt;</td> </tr> <tr> <td>Over Draft&lt;/td&gt;</td> <td>A, B&lt;/td&gt;</td> <td>Home Loan,Personal Loan,Over Draft&lt;/td&gt;</td> </tr> </table> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO, , submissionInputDTO, , Indicates the draft submission input DTO @return StructureSolutionFactsResponse, An instance of the response containing facts per lending product group required for structure solution. @throws FatalException</p>	<b>Party</b></td>	<b>Product Group</b></td>	<b>Requested Amount</b></td>	A</td>	Home Loan</td>	100k</td>	B</td>	Personal Loan</td>	50k</td>	A, B</td>	Over Draft</td>	150k</td>	<b>For Product Group</b></td>	<b>Party</b></td>	<b>Facility</b></td>	Home Loan</td>	A</td>	Home Loan, Over Draft</td>	Personal Loan</td>	B</td>	Personal Loan, Over Draft</td>	Over Draft</td>	A, B</td>	Home Loan,Personal Loan,Over Draft</td>
<b>Party</b></td>	<b>Product Group</b></td>	<b>Requested Amount</b></td>																										
A</td>	Home Loan</td>	100k</td>																										
B</td>	Personal Loan</td>	50k</td>																										
A, B</td>	Over Draft</td>	150k</td>																										
<b>For Product Group</b></td>	<b>Party</b></td>	<b>Facility</b></td>																										
Home Loan</td>	A</td>	Home Loan, Over Draft</td>																										
Personal Loan</td>	B</td>	Personal Loan, Over Draft</td>																										
Over Draft</td>	A, B</td>	Home Loan,Personal Loan,Over Draft</td>																										

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchDirectExposureDetails		This service will return facts required for pricing during structure solution in pre assessment for all lending product groups in the submission. Currently total direct credit exposure is being populated. It is computed in LCM for existing parties and proposed and credit facilities.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the submission identifier @return StructureSolutionFactsResponse, An instance of the response containing facts per lending product group required for structure solution. @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchDirectExposureDetails		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	deleteValuationRequestDraftForExtRefAndCollateralID		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	deleteValuationRequestDraftForExtRefAndCollateralID		This service is called by the collateral valuation task to update the state of the collateral valuation Task in Origination. Each CollateralDecision entity stores the ValuationStateType and TitleSearchStateType. If the ValuationStateType is COMPLETED Or WAIVED and TitleSearchStateType is COMPLETED, then this service will update the EvaluationStateType that is also stored at CollateralDecision. Each

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ApplicationDecision entity stores the EvaluationStateType that is updated by this service when all the CollateralDecision within ApplicationDecision has EvaluationStateType as COMPLETED. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @param String,applicationId, ,applicationId Unique Identifier for the application @param ValuationStateType,valuationStateType, ,valuationStateType This attribute indicates the state of valuation task of the collateral @param CollateralApprovalDataDTO, ,collateralApprovalDataDTO, ,collateralApprovalDataDTO This attribute indicates the collateral level approval data DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_VALUATION_STATE_INCORRECT} : FC_OR_CD_0021&lt;br&gt; This exception is thrown if valuation state type in the call to this service does not equal COMPLETED&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteValuationRequestDraftForExtRefAndCollateralID		<p>This service is called by the collateral valuation task to update the state of the collateral valuation Task in Origination. Each CollateralDecision entity stores the ValuationStateType and TitleSearchStateType. If the ValuationStateType is COMPLETED Or WAIVED and TitleSearchStateType is COMPLETED, then this service will update the EvaluationStateType that is also stored at CollateralDecision. Each ApplicationDecision entity stores the EvaluationStateType that is updated by this service when all the CollateralDecision within ApplicationDecision has EvaluationStateType as COMPLETED. @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @param String,applicationId, ,applicationId Unique Identifier for the application @param ValuationStateType,valuationStateType, ,valuationStateType This attribute indicates the state of valuation task of the collateral @param CollateralApprovalDataDTO, , collateralApprovalDataDTO, , collateralApprovalDataDTO This attribute indicates the collateral level approval data DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				found for the given submission id.  {@link OriginationErrorConstants#OR_VALUATION_STATE_INCORRECT} : FC_OR_CD_0021  This exception is thrown if valuation state type in the call to this service does not equal WAIVED  {@link OriginationErrorConstants#OR_WAIVE_VALUATION_APPROVAL_REQUIRED} : FC_OR_CD_0022  This error is raised to send the collateral valuation waiver for DCA approval based on rule configuration  @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteValuationRequestDraftForExtRefAndCollateralID		
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteTitleSearchRequestDraftForExtRefAndCollateralID		
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	deleteTitleSearchRequestDraftForExtRefAndCollateralID		This service is called by the collateral title search task to update the state of the collateral title search Task in Origination. Each CollateralDecision entity stores the ValuationStateType and TitleSearchStateType. If the ValuationStateType is COMPLETED Or WAIVED and TitleSearchStateType is COMPLETED, then this service will update the EvaluationStateType that is also stored at CollateralDecision. Each ApplicationDecision entity stores the EvaluationStateType that is updated by this service when all the CollateralDecision within ApplicationDecision has EvaluationStateType as COMPLETED. @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc  @param String,submissionId, ,submissionId Unique Identifier for the submission @param String,applicationId, ,applicationId Unique Identifier for the application @param TitleSearchStateType,titleSearchStateType, ,titleSearchStateType This attribute indicates the state of title search task of the collateral @param CollateralApprovalDataDTO, , collateralApprovalDataDTO, , collateralApprovalDataDTO This attribute indicates the collateral level approval data DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_WAIVE_TITLE_SEARCH_APPROVAL_REQUIRED} : FC_OR_CD_0024&lt;br&gt; This error is raised to send the collateral title search waiver for DCA approval based on rule configuration&lt;br&gt; @throws FatalException</p>
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	deleteTitleSearchRequestDraftForExtRefAndCollateralIID		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	calculateUniformLTVForProjectedCreditFacility		
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	confirmBorrowingEntity		This service will pass a list of party identifiers and fetch the borrowing entity. If the borrowing entity exists, the same is passed else a new borrowing entity is created and passed.  Borrowing entities are created and maintained in LCM system. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Map<Integer,ApplicantDataDTO>,applicantMap,, Indicates a map of applicants with applicant serial number as the key @fcb.param MI,ProductDetailsDTO,productDetailsDTO,, Indicates the product details. @return BorrowingEntityInquiryResponse,, Contains the borrowing entity information created @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchFacilityCategory		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityCategory		This service is used for defaulting facility category related fields in credit facility.  In Origination, only basic fields such as requested amount and term for facility, purpose  and variant information such as employee benefits required, whether facility is low doc  and capital subvention scheme is being used for the facility.    Facility Category for the product group under which a facility is being setup is inquired  from PMU. Facility Category fields are further fetched using this facility category code  and fields are defaulted to the facility being setup. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategory, , Indicates the facility category code @return CreditFacilityDTO, Returns the credit facility dto populated with default category fields @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityCategory		This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	fetchFacilityCategory		Fetch Facility Category based on facility category code @param sessionContext @param facilityCategoryId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	generateFacilityId		<p>This service will generate the facility Id and default facility category fields. In Origination, only basic fields such as requested amount and term for facility, purpose&lt;br&gt; and variant information such as employee benefits required, whether facility is low doc&lt;br&gt; and capital subvention scheme is being used for the facility.&lt;br&gt; &lt;br&gt; Facility Category for the product group under which a facility is being setup is inquired&lt;br&gt; from PMU. Facility Category fields are further fetched using this facility category code&lt;br&gt; and fields are defaulted to the facility being setup.&lt;br&gt; &lt;br&gt; The next available facility identifier is generated in LCM system and fetched. The &lt;br&gt; facility will be created using this identifier. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ProductGroupLinkageDTO, productGroupLinkageDTO, , This contains the product group details for which facility has been requested @return CreditFacilityResponse, Returns the credit facility dto populated with default category fields and facility identifier @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_CANNOT_BE_SETUP} : FC_OR_DE_0077 &lt;br&gt; This error occurs when:&lt;br&gt; Facility is being setup for product groups that are not of the type - Assets.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FACILITY_ID_NOT_GENERATED} : FC_OR_DE_0078 &lt;br&gt; This error occurs</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				when:  Facility Identifier generation fails or no identifier is returned by LCM service.    {@link OriginErrorConstants#FACILITY_CATEGORY_INVALID} : FC_OR_DE_0083   This error occurs when:  Facility Category Identifier is incorrect or no details are returned from LCM service. 
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateFacilityId		This method would generate the new facility-id for the given {@link CreditFacilityDTO} @param creditFacilityDTO, instance of {@link CreditFacilityDTO} @return Facility Id @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	generateFacilityId		This method would generate the new facility-id for the given {@link CreditFacilityDTO} @param creditFacilityDTO, instance of {@link CreditFacilityDTO} @return Facility Id @throws FatalException
IOriginationLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAdapter	fetchProposedFacilityForCollateral		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	isManualAllocationRequired	This method will call the <a href="#">{@link LTVApplicationService#isManualAllocationRequired(SessionContext, CreditFacilityDTO[])}</a> .  This method indicates whether the Uniform LVR Allocation details should be shown depending on some parameter changes like collateral value changes etc @param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException	@param sessionContext @param submissionId @return @throws FatalException
IOriginationLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAadapter	isManualAllocationRequired	This method will call the <a href="#">{@link LTVApplicationService#isManualAllocationRequired(SessionContext, CreditFacilityDTO[])}</a> .  This method indicates whether the Uniform LVR Allocation details should be shown depending on some parameter changes like collateral value changes etc @param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException	This method performs the validations listed below: <li>Rejected product groups - if any of the product group related offer document is rejected, the product group is marked as 'Offer_REJECTED' and withdrawn. <li>LTV code deterioration - If LTV has deteriorated for any product group , Collateral valuation and Collateral title search @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,Set<OfferDocumentLinkage> all the offer documents for the application @return List<ValidationError> if any

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	isManualAllocationRequired	This method will call the <code>@link LTVApplicationService#isManualAllocationRequired(SessionContext, CreditFacilityDTO[])</code> . This method indicates whether the Uniform LVR Allocation details should be shown depending on some parameter changes like collateral value changes etc @param sessionContext @param submissionID @param proposedFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This service checks for collateral valuation completion, LVR and LTV code deterioration based on the following cases: If NCCP = yes and LRM condition code does not exist or is waived, valuation must be completed for all collateral and LTV code deterioration check must be done. If NCCP = yes and LRM condition code exists, valuation must be completed for all collateral, LVR threshold breach must be checked and LTV code deterioration check must be done. If NCCP = no and LRM condition code exists, valuation must be completed for all collateral and LVR threshold breach must be checked. If NCCP = no and LRM condition code does not exist or is waived, no check is required. The LRM_CONDITION_CODE is configured in the OriginationConfiguration properties file. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, ,lendingProductGroupLinkageDTO, , Indicates the lending product group for which LTV deterioration is being verified. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR when compared to the LVRs present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This method checks whether there is any deterioration in the latest LVR when compared to the LVRs present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRValueDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR value when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This method validates whether the LVR value has been changed from the last time value. If no change in LVR value then it return false otherwise returns true. @param sessionContext @param lendingPgLinkageDTO @param collaterals @param applicantDatas @param creditFacilityDTOs @return Boolean
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRValueDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR value when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This method checks whether there is any deterioration in the latest LVR value when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRValueDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR value when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This service check if the current LVR has deteriorated beyond the threshold values permissible at the product group level. If yes, return an error.  The previous LVR values are retrieved from the LTV history maintained at the proposed facility level.  A call to calculate the LVR is again made and value is retrieved.  The difference between the new LVR and old LVR value retrieved from history are compared to the threshold value  maintained at the Credit Policy Template level for the given product group.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityId, , Indicates the facility identifier for which LVR deterioration is being verified. @fcb.param MI,String, productGroupId, , Indicates the product group for which facility has been configured. The threshold value is inquired from the Credit policy template maintained for this product group. @return ValidationError, Error is thrown if the LVR has deteriorated beyond threshold defined at Product group level @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRCodeDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR code (LCC) when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This method checks whether there is any deterioration in the latest LVR code (LCC) when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	performLVRCodeDeteriorationCheck	This method checks whether there is any deterioration in the latest LVR code (LCC) when compared to the LVR s present at credit decision level (Decision LVR) @param sessionContext @param projectedCreditFacilityIds @return @throws com.ofss.fc.infra.exception.FatalException	This method calculates the LTV deterioration for a given facility ID. The LTV calculated during the credit assessment stage and the latest LTV value is compared and a change in the Lending category code(LCC) is checked.  A validation error is throws if the LTV is deteriorated and a mismatch in the LCC is found i.e. the previous LCC is not equal to the latest calculated LCC.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityId, , Indicates the facility identifier for which LVR deterioration is being verified. @fcb.param MI,String, productGroupName, , Indicates the name of the product group to be displayed in case of LTV deteriorated for that product group @fcb.param MI,Date, startDate, , Indicates the start date for range for which LTV history is to be retrieved @fcb.param MI,Date, endDate, , Indicates the end date for range upto which LTV history is to be retrieved @return ValidationError, Return an error if LTV has deteriorated. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	checkManualAllocationTaskStatus	@param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException	This method performs the validations listed below: <li>Rejected product groups - if any of the product group related offer docuemnt is rejected, the product group is marked as 'Offer_REJECTED' and withdrawn. <li>LTV code deterioration - If LTV has deteriorated for any product group , Collateral valuation and Collateral title search @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,String applicationId , Indicates the lending application for which the create offer stage is in progress @fcb.param MI,Set<OfferDocumentLinkage> all the offer documents for the application @return List<ValidationError> if any
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	checkManualAllocationTaskStatus	@param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException	@param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	checkManualAllocationTaskStatus	@param sessionContext @param submissionID @param proposedFacilityIDs @return @throws com.ofss.fc.infra.exception.FatalException	This service checks for collateral valuation completion, LVR and LTV code deterioration   based on the following cases:  If NCCP = yes and LRM condition code does not exist or is waived, valuation must be completed for all  collateral and LTV code deterioration check must be done.  If NCCP = yes and LRM condition code exists, valuation must be completed for all collateral, LVR   threshold breach must be checked and LTV code deterioration check must be done.  If NCCP = no and LRM condition code exists, valuation must be completed for all collateral and LVR   threshold breach must be checked.  If NCCP = no and LRM condition code does not exist or is waived, no check is required.  The LRM_CONDITION_CODE is configured in the OriginationConfiguration properties file.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, ,lendingProductGroupLinkageDTO, , Indicates the lending product group for which LTV deterioration is being verified. @throws FatalException
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	saveFacilityChargeGraphsAndUpdateDecisionLVR	This method is used to persist the Facility Charge Graphs along with updating decision details for the given proposed facility Ids which are part of the submission	This method is used to persist the Facility Charge Graphs along with updating decision details for the given proposed facility Ids which are part of the submission

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	saveGraphsAndUpdateLVRs	This method is used to save the facility charge graphs. It retains any manual allocations if present (applicable only to Uniform method) and a fresh a calculation is done otherwise It also updates facility LVR Details (all proposed facilities in the group) and Facility decision details (proposed facilities in sub,ission only)	This method is used to save the facility charge graphs. It retains any manual allocations if present (applicable only to Uniform method) and a fresh a calculation is done otherwise It also updates facility LVR Details (all proposed facilities in the group) and Facility decision details (proposed facilities in sub,ission only)
IOriginationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationLCMAAdapter	saveGraphsAndUpdateLVRs	This method is used to save the facility charge graphs. It retains any manual allocations if present (applicable only to Uniform method) and a fresh a calculation is done otherwise It also updates facility LVR Details (all proposed facilities in the group) and Facility decision details (proposed facilities in sub,ission only)	This service is used to create new facilities or update existing facilities.  The following activities will be carried out in sequence.    a) Make a call to LCM system to check if a borrowing entity exists for the parties applying for the  facility. LCM internally checks if an existing borrowing entity exists for the parties passed and   returns. Otherwise, LCM creates a new borrowing entity and returns the newly created borrowing entity.   b) Update the borrowing entity identifier against the facility.   c) Make a call to LCM system to create or amend existing proposed facility. LCM internally creates   linkages to collaterals linked to the facility. Also, simulations that are attached to the facility  either to the main facility or sub-facility (whichever exists at lowest level) are stored. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO,submissionInputDTO,, Indicates the submission details containing the applicant and facility information. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>required information about the status of execution indicating if the invocation succeeded or failed. &lt;br&gt;  @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#BORROWING_ENTITY_ID_CREATE_FAILED} : FC_OR_DE_0076 &lt;br&gt; This error occurs when:&lt;br&gt; The borrowing entity identifier for the group of applicants applying for a facility could not be generated.&lt;br&gt;</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDailyInterestComputation	<p>This method will be called to invoke interest computation for an account for a process date. Balances will be passed to the service on which the interest will be calculated. DDA also pass the information about the state of the DemandDepositAccount, whether it is liable for credit interest or debit interest. @param accountId Account Id for which the interest computation is invoked. @param balancesMap balances for which the interest will be computed. @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed.</p>	<p>This method will be called during closure of a CASA account. The method will update the interest plan with the closure date which will be the process date. The method in turn calls {@link InterestOperationService#doInterestProcessingOnAccountClosure} @param sessionContext @param accountNumber @param depositNumber @param closureDate @param productClass</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDailyInterestComputation	<p>This method will be called to invoke interest computation for an account for a process date. Balances will be passed to the service on which the interest will be calculated. DDA also pass the information about the state of the DemandDepositAccount, whether it is liable for credit interest or debit interest. @param accountId Account Id for which the interest computation is invoked. @param balancesMap balances for which the interest will be computed. @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDailyInterestComputation	<p>This method will be called to invoke interest computation for an account for a process date. Balances will be passed to the service on which the interest will be calculated. DDA also pass the information about the state of the DemandDepositAccount, whether it is liable for credit interest or debit interest. @param accountId Account Id for which the interest computation is invoked. @param balancesMap balances for which the interest will be computed. @param requestDate date on which the interest computation is requested. @param effectiveDate effective date for the interest computation. @param isCredit flag to indicate whether credit interest is computed or debit interest is computed.</p>	<p>This method calls the interest module to compute the interest, It may be called at the time of account opening or at the time of online balance change or at the time of batch processing. &lt;br&gt; So the same method is called at each time to interact with interest module to compute the interest for the account.&lt;br&gt;This method will also be called for computing back dated interest. This method will be designed in such a manner so that all calls to interest computation will only occur via this method only.&lt;br&gt;CASA module need to send few details to interest on the basis of which Interest module compute the interest for an account.&lt;br&gt; &lt;ul&gt; &lt;li&gt;arrangement number : account number for which the interest will be computed. &lt;li&gt;interest class : CASA module can easily predict the state of an account, i.e. whether the account is in Credit state of Debit state.&lt;br&gt; Bank accounts may have a positive, or credit balance, where the bank owes money to the customer; or a negative, or debit balance, where the customer owes the bank money.&lt;br&gt; &lt;ol&gt; &lt;li&gt;Credit : This states that a CASA account has positive balance. In such case, bank owes money to the customer. In banking terms, customer din't utilised neither of the OD, TOD, Overline and AUF facility.&lt;br&gt; &lt;li&gt;Debit : This states that a CASA account has negative balance. In such case, customer owes money to the bank. In banking terms, customer has utilised either of the OD, TOD, Overline or AUF facility to draw funds.&lt;br&gt; &lt;/ol&gt; &lt;li&gt;interest division : Interest can either be calculated on the regular amounts, or can be calculated on the bonus. For calculation of interest in different amount types, different rules are defined. So, the interest module&lt;br&gt; will be intimated about the interest division. For CASA it will be always Regular. For</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>more Details on this see <a href="#">{@link InterestDetails}</a></p> <ul style="list-style-type: none"> <li>&lt;li&gt;product class : Interest module need to know about the calling module. For CASA it will be <a href="#">{@link ProductClass#CASA}</a></li> <li>&lt;li&gt;request date : Dates are very important in Banking. Request date will always be the process Date when the interest module is invoked for interest computation.</li> <li>&lt;li&gt;effective date : When interest is computed on a balance change event, effective date will be same as of request date. But when the interest will be computed for a back dated entry&lt;br&gt; interest will be computed from that date, thus effective date will be that back dated date.&lt;br&gt; Effective date can either be equal to request date or less than the request date, but it cannot be greater than the request date.&lt;br&gt;</li> <li>&lt;li&gt;Map of balances : Interest will be calculated on some balance amounts. There are different balances in CASA module, and each balance may have a different rate of interest.&lt;br&gt;</li> <li>&lt;ol&gt;</li> <li>&lt;li&gt;Clear Balance : Cash deposited by the customer into the bank. For more see <a href="#">{@link EffectiveBalances#clearBalance}</a></li> <li>&lt;li&gt;Unclear Balance : Cash deposited by the customer in form of cheque or other instrument which is awaiting confirmation. For more see <a href="#">{@link EffectiveBalances#unclearBalance}</a></li> <li>&lt;li&gt;Overdraft Utilisation : Cash overdrawn from the account by using Overdraft facility. For more see <a href="#">{@link EffectiveBalances#utilizedOverdraftLimit}</a>.</li> <li>&lt;li&gt;Overline Utilisation : Cash overdrawn from the account by using Overline facility above overdraft utilisation. For more see <a href="#">{@link EffectiveBalances#overlineAmount}</a>.</li> <li>&lt;li&gt;TOD Utilisation : Cash overdrawn from the account by using Temporary Overdraft facility. For more see <a href="#">{@link EffectiveBalances#todAmount}</a>.</li> <li>&lt;li&gt;AUF Utilisation : Cash overdrawn from the account by</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>using Advance Against Unclear Funds facility. For more see {@link EffectiveBalances#utilizedAUFAmount}. &lt;/ol&gt; &lt;/ul&gt;This method works as follows : &lt;br&gt; &lt;pre&gt; requestDate = processDate effectiveDate = effectiveDate First check request date &amp;gt;= effective date. If the above condition satisfies, then determine whether account is in credit state or debit state. If any one of AUF, OD, TOD, Overline amounts is not zero, then the account is in debit state, else it is in credit state. To check this: set isCredit = true; for (Map.Entry&lt;BalanceType, Money&gt;; balanceMap: queryMap.entrySet()) { if(balanceMap.getKey() == BalanceType#AUFUtilized    balanceMap.getKey() == BalanceType#OverdraftUtilized    balanceMap.getKey() == BalanceType#TODUtilized    balanceMap.getKey() == BalanceType#Overline) { if(balanceMap.getValue() &amp;gt; 0)&amp;lt;i&amp;gt;0 refers to zero money of the same currency as of account&amp;lt;/i&amp;gt; &amp;lt;i&amp;gt;currency to make zeroMoney can be retrieved from any value of balanceMap object&amp;lt;/i&amp;gt; { set isCredit = false; break; } } } Since, interest computation need to know the state of account, to compute either credit or debit interest.&amp;lt;br&amp;gt; DDA module can easily predict the state of the account, by analyzing balances.&amp;lt;br&amp;gt; Call Interest adapter to invoke interest computation. Call {@link IDDAInterestAdapter#doDailyInterestComputation()} &lt;/pre&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	inquireInterestDetails	This method is used to fetch credit and debit interest information for demand deposit account.Method provides following information for credit and debit interest. <ol><li>Interest accrual information: Frequency,next accrual date,accrual basis. <li>Interest capitalization information: Frequency,next accrual date,accrual basis. <li>Interest settlement statistics: Interest paid/received till date. </ol> If Account is opened under liability product then all debit interest information is 0 always. @param sessionContext @param accountId	This method is used to fetch credit and debit interest information for demand deposit account.Method provides following information for credit and debit interest. <ol><li>Interest accrual information: Frequency,next accrual date,accrual basis. <li>Interest capitalization information: Frequency,next accrual date,accrual basis. <li>Interest settlement statistics: Interest paid/received till date. </ol> If Account is opened under liability product then all debit interest information is 0 always. TODO: inquiry method in interest is not available as of now.I have raised request for the same . fetchInterestPlanItemDetails method is called twice one by passing Credit and another by passing Debit. TODO TODO @param accountId - This is the unique identifier for an account. @param sessionContext returns InterestInquiryDetailsDTO @throws FatalException
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	inquireInterestDetails	This method is used to fetch credit and debit interest information for demand deposit account.Method provides following information for credit and debit interest. <ol><li>Interest accrual information: Frequency,next accrual date,accrual basis. <li>Interest capitalization information: Frequency,next accrual date,accrual basis. <li>Interest settlement statistics: Interest paid/received till date. </ol> If Account is opened under liability product then all debit interest information is 0 always. @param sessionContext @param accountId	This method provides credit and debit interest details for demand deposit account. Method provides following types of interest information. <ol><li>Interest accrual information: Frequency,next accrual date,accrual basis. <li>Interest capitalization information: Frequency,next accrual date,accrual basis. <li>Interest settlement statistics: Interest paid/received till date. </ol> There are two kinds of Demand Deposit products, Asset and Liability. Accounts opened under a liability type of product cannot go into negative. Hence only credit interest is applicable for such accounts. Accounts opened under an asset type of product can go into negative, hence both debit and credit interests are applicable for such accounts.   Interest information is stored in interest module.This method calls {@link IDDAInterestAdapter} to get information. Some of the important points about balance details mentioned



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>below</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;All interests amounts are expressed in account currency only.</li> <li>&lt;li&gt;All amounts are equal or greater than zero.</li> </ul> </li> </ul> <p>&lt;/ol&gt;</p> <p>&lt;br&gt;</p> <p>This method performs following activities.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Validates demand deposit account.It calls{@link DemandDepositAccountService#validateDemandDepositAccount}.</li> <li>&lt;li&gt;Calls {@link IDDAInterestAdapter#inquireInterestDetails}.</li> <li>&lt;li&gt;Prepare response object.</li> </ul> </li> </ul> <p>&lt;/ol&gt;</p> <p>@fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId, &lt;br&gt; The attribute required to fetch interest details of DDAccount. @return {@link InterestInquiryDetailsResponse} &lt;br&gt; InterestInquiryDetailsResponse contains InterestInquiryDetailsDTO which will contain following information.&lt;br&gt;</p> <p>&lt;b&gt;creditInterestCapitalisationFrequency&lt;/b&gt;&lt;/p&gt; <p>This attribute denotes credit interest capitalization frequency like Monthly,Quarterly etc.&lt;br&gt;</p> <p>&lt;b&gt;debitInterestCapitalisationFrequency&lt;/b&gt;&lt;/p&gt; <p>This attribute denotes credit interest capitalization frequency like Monthly,Quarterly etc.&lt;br&gt;</p> <p>&lt;b&gt;nextCreditInterestCapitalizationDate&lt;/b&gt;&lt;/p&gt; <p>This attribute denotes next date for credit interest capitalization.&lt;br&gt;</p> <p>&lt;b&gt;creditInterestCapitalisationBasis&lt;/b&gt;&lt;/p&gt; <p>This attribute denotes day count convention basis for credit interest capitalization.&lt;br&gt;</p> <p>&lt;b&gt;debitInterestCapitalisationBasis&lt;/b&gt;&lt;/p&gt; <p>This attribute denotes day count convention basis for debit interest capitalization.&lt;br&gt;</p> <p>&lt;b&gt;nextDebitInterestCapitalizationDate&lt;/b&gt;&lt;/p&gt; <p>This</p> </p></p></p></p></p></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>attribute denotes next date for debit interest capitalization. &lt;br&gt;  &lt;b&gt;creditInterestAccruedTillDate&lt;/b&gt; This field denotes credit interest accrued in the account after the last capitalization.&lt;br&gt;  &lt;b&gt;debitInterestAccruedTillDate&lt;/b&gt; This field denotes debit interest accrued in the account after the last capitalization. &lt;br&gt;  &lt;b&gt;creditInterestPaidYearTillDate&lt;/b&gt; This attribute denotes credit interest settled in the account for the current financial year.&lt;br&gt;  &lt;b&gt;debitInterestPaidYearTillDate&lt;/b&gt; This attribute denotes debit interest settled in the account for the current financial year.&lt;br&gt; funds.&lt;br&gt; In case of failure -&lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; &lt;ul&gt;  &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	unregisterPlan	This method will be used to unregister an interest plan. If the account is being delinked or the interest plan of an account is being changed, the existing plan needs to be phased out.   @param accountId @param depositNumber	This method will be used to unregister an interest plan. If the account is being delinked or the interest plan of an account is being changed, the existing plan needs to be phased out.  The method will in turn call the service {@link InterestPlanService#unregisterInterestPlan} @param accountId @param depositNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	unregisterPlan	This method will be used to unregister an interest plan. If the account is being delinked or the interest plan of an account is being changed, the existing plan needs to be phased out.   @param accountId @param depositNumber	This method is common activity which can be called by any service for activities such as 1.validate & delink bundle benefits from the account to be closed by call to bundle 2.close pricing plan by calling pricing 3.close interest plan by calling interest 4.unregister taxation plan by calling taxation 5.remove all the actions on the account 6.Remove all proportioned OD logged from effective date. 7.delink alternate account. @param closureDTO @param account @param agingAccountStatus @param finalBalance @throws FatalException
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	fetchDetailsOnCreditInterestSettlement	Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount.<p>This method is used to fetch details on credit interest settlement action.</p> @see InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed	Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount. <p>The method works as follows : <ol><li>Prepare the InterestSettlementRequest as follows : InterestSettlementRequest#arrangementNumber = input.accountId. InterestSettlementRequest#interestClasDates = Credit InterestSettlementRequest#interestDivision = Regular InterestSettlementRequest#requestDate = processDate InterestSettlementRequest#dueDate = input.settlementDate InterestSettlementRequest#isArrangementBeingClosed = input.isAccountBeingClosed InterestSettlementRequest#transactionReferenceNumber = TODO</li> <li>Call the service {@link InterestSettlementService#doInterestSettlement} to fetch the credit interest settlement details.</li> <li>Populate the details returned by the service in InterestSettlementDetailsDTO and return.</li> </p> @see

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.domain.interest.service.InterestSettlementService @see com.ofss.fc.domain.interest.service.InterestSettlementService @param accountId -Identifies Account. @param settlementDate @param isAccountBeingClosed
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	fetchDetailsOnCreditInterestSettlement	Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount.<p>This method is used to fetch details on credit interest settlement action.</p> @see InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed	Please refer class level documentation of this class.  Bank gives credit interest on the money deposited by the customer. This process is called Credit interest settlement. The settlement takes place at a certain frequency called as the settlement frequency. The frequency is stored in the Interest module as part of Interest Rule Definition.  This method is used to perform credit interest settlement. As a result of this process, event is logged for passing accounting entries, the interest amount is credited to the account or to an alternate account and a future dated action is logged for the next settlement date.  In case of alternate account ,The {@code creditInterestAmount} will credit interest in the alternate account, action for next settlement date will be logged for the parent account. This method performs following steps <ol> <li>Load(validate) demand deposit account using {@link DemandDepositAccountService#fetchDemandDepositAccount}. <li>Get Settlement amount :Call {@link IDDAInterestAdapter#fetchDetailsOnCreditInterestSettlement} <li>If Settlement amount > 0 then <ol> <li>Get alternate account for credit interest posting. {@code fetchAlternateAccount}. <li>Validate Business policy. <ol> <li>Prepare {@link InterestSettlementBusinessPolicyDTO}. <li>Call {@link InterestSettlementBusinessPolicy}. </ol></li><li>Credit interest amount: Call {@code

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>creditInterestAmount} method. &lt;li&gt;Raise Accounting event : Call {@code raiseAccountingEvent} method. &lt;li&gt;Log action for next credit interest settlement date. Call {@link ActionLogger.logDDAction(DDActionType#CreditInterestSettlement)} &lt;/ol&gt; &lt;/ol&gt; TODO: 1) Ask FS what will happen if status is {@link DemandDepositLifeCycleStatus#creditBlocked}. Raise jira for it.2) Ask FS about what is credit alternate account is Loan or TD. Need IMC for these conditions. @param applicationContext Application Context which will be set in batch process. @param accountId DDAccountId for which settlement process is being done. @param settlementDate credit interest settlement date @param isAccountClosed flag to check whether account is closed or not. @return settlementAmount @throws, method can throw following unchecked exception &lt;ol&gt; &lt;li&gt; {@link DemandDepositAccountErrorConstant#DDA_INVALID_ACCOUNT_STATUS}. &lt;li&gt; {@link DemandDepositAccountErrorConstant#DDA_BLANK_INPUT}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	fetchDetailsOnCreditInterestSettlement	<p>Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount.&lt;p&gt;This method is used to fetch details on credit interest settlement action.&lt;/p&gt; @see InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed</p>	<p>Please refer class level documentation of this class.</p> <p>&lt;br&gt; In liability type of product, Customer can overdraw funds using following facilities &lt;ol&gt; &lt;li&gt;Overdraft facility- Please refer {@link EffectiveBalances#utilizedOverdraftLimit} and OverdraftLimit &lt;li&gt;Overline facility Please refer {@link EffectiveBalances#overlineAmount} &lt;li&gt;TOD facility. Please refer {@link EffectiveBalances#todAmount} &lt;/ol&gt; Bank levies heavy interest on money which is overdrawn. This charging process is called as debit interest settlement. The settlement takes place at a certain frequency called as the settlement frequency. The frequency is stored in the Interest module as part of Interest Rule Definition.</p> <p>&lt;br&gt;This method is used to perform debit interest settlement. As a result of this process, event is logged for passing accounting entries, the interest amount is debited from the account or from an alternate account and a future dated action is logged for the next settlement date. &lt;br&gt; In case of alternate account ,The {@code debitInterestAmount} will debit interest amount from alternate account, action for next settlement date will be logged for the parent account. This method performs following steps &lt;ol&gt; &lt;li&gt;Load(validate) demand deposit account using {@link DemandDepositAccountService#fetchDemandDepositAccount}. &lt;li&gt;Get Settlement amount :Call {@link IDDAInterestAdapter#fetchDetailsOnDebitInterestSettlement} &lt;li&gt;If Settlement amount &gt; 0 then &lt;ol&gt; &lt;li&gt;Get alternate account for debit interest posting.In case of debit interest settlement, {@code fetchAlternateAccount}. &lt;li&gt;Validate Business policy. &lt;ol&gt; &lt;li&gt;Prepare {@link InterestSettlementBusinessPolicyDTO}. &lt;li&gt;Call</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link InterestSettlementBusinessPolicy}. &lt;/ol&gt;  &lt;li&gt;Debit interest amount: Call {@code debitInterestAmount} method. &lt;li&gt;Raise Accounting event : Call {@code raiseAccountingEvent} method.  &lt;li&gt;Log action for next debit interest settlement date. Call {@link ActionLogger.logDDAction(DDActionType#DebitInterestSettlement)} &lt;/ol&gt; &lt;/ol&gt; @param applicationContext Application Context which will be set in batch process. @param accountId DDAccountId for which settlement process is being done. @param settlementDate debit interest settlement date @param isAccountClosed flag to check whether account is closed or not. @return settlementAmount @throws, method can throw following unchecked exception  &lt;ol&gt; &lt;li&gt; {@link DemandDepositAccountErrorConstant#DDA_INVALID_ACCOUNT_STATUS}. &lt;li&gt; {@link DemandDepositAccountErrorConstant#DDA_BLANK_INPUT}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	fetchDetails OnDebitInterestSettlement	<p>Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount.&lt;p&gt;This method is used to fetch details on debit interest settlement action.</p> <p>&lt;/p&gt; @see InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed</p>	<p>Interest module is responsible to calculate credit or debit interest settlement amount. On debit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount. &lt;p&gt;The method works as follows : &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Prepare the InterestSettlementRequest as follows : InterestSettlementRequest#arrangementNumber = input.accountId.</li> <li>InterestSettlementRequest#interestClasDates = Debit</li> <li>InterestSettlementRequest#interestDivision = Regular</li> <li>InterestSettlementRequest#requestDate = processDate</li> <li>InterestSettlementRequest#dueDate = input.settlementDate</li> <li>InterestSettlementRequest#isArrangementBeingClosed = input.isAccountBeingClosed</li> <li>InterestSettlementRequest#transactionReferenceNumber = TODO&lt;/li&gt;</li> <li>&lt;li&gt;Call the service {@link InterestSettlementService#doInterestSettlement} to fetch the credit interest settlement details.&lt;/li&gt;</li> <li>&lt;li&gt;Populate the details returned by the service in InterestSettlementDetailsDTO and return.&lt;/li&gt;</li> </ul> <p>&lt;/p&gt; @see com.ofss.fc.domain.interest.service.InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	fetchDetailsOnDebitInterestSettlement	<p>Interest module is responsible to calculate credit or debit interest settlement amount. On credit interest settlement event,demand deposit module calls interest module to get settlement amount.In response to this request interest module gives next settlement date as per settlement frequency along with settlement amount.&lt;p&gt;This method is used to fetch details on debit interest settlement action.&lt;/p&gt; @see InterestSettlementService @param accountId @param settlementDate @param isAccountBeingClosed</p>	<p>Please refer class level documentation of this class.</p> <p>&lt;br&gt; In liability type of product, Customer can overdraw funds using following facilities &lt;ol&gt; &lt;li&gt;Overdraft facility- Please refer {@link EffectiveBalances#utilizedOverdraftLimit} and OverdraftLimit &lt;li&gt;Overline facility Please refer {@link EffectiveBalances#overlineAmount} &lt;li&gt;TOD facility. Please refer {@link EffectiveBalances#todAmount} &lt;/ol&gt; Bank levies heavy interest on money which is overdrawn. This charging process is called as debit interest settlement. The settlement takes place at a certain frequency called as the settlement frequency. The frequency is stored in the Interest module as part of Interest Rule Definition.</p> <p>&lt;br&gt;This method is used to perform debit interest settlement. As a result of this process, event is logged for passing accounting entries, the interest amount is debited from the account or from an alternate account and a future dated action is logged for the next settlement date. &lt;br&gt; In case of alternate account ,The {@code debitInterestAmount} will debit interest amount from alternate account, action for next settlement date will be logged for the parent account. This method performs following steps &lt;ol&gt; &lt;li&gt;Load(validate) demand deposit account using {@link DemandDepositAccountService#fetchDemandDepositAccount}. &lt;li&gt;Get Settlement amount :Call {@link IDDAInterestAdapter#fetchDetailsOnDebitInterestSettlement} &lt;li&gt;If Settlement amount &gt; 0 then &lt;ol&gt; &lt;li&gt;Get alternate account for debit interest posting.In case of debit interest settlement, {@code fetchAlternateAccount}. &lt;li&gt;Validate Business policy. &lt;ol&gt; &lt;li&gt;Prepare {@link InterestSettlementBusinessPolicyDTO}. &lt;li&gt;Call</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link InterestSettlementBusinessPolicy}. &lt;/ol&gt;  &lt;li&gt;Debit interest amount: Call {@code debitInterestAmount} method. &lt;li&gt;Raise Accounting event : Call {@code raiseAccountingEvent} method.  &lt;li&gt;Log action for next debit interest settlement date. Call {@link ActionLogger.logDDAction(DDActionType#DebitInterestSettlement)} &lt;/ol&gt; &lt;/ol&gt; @param applicationContext Application Context which will be set in batch process. @param accountId DDAccountId for which settlement process is being done. @param settlementDate debit interest settlement date @param isAccountClosed flag to check whether account is closed or not. @return settlementAmount @throws, method can throw following unchecked exception  &lt;ol&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_INVALID_ACCOUNT_STATUS}. &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_BLANK_INPUT}</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer {@link IDDAInterestAdapter#registerInterestPlan} @param IDDAInterestRegistrationDTO	<p>The method is responsible for registering an interest plan for the account in the interest module at the time of account opening.This method is referred from {@link DDAccountOpeningService#openDemandDepositAccount()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse } to register in interest module. InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product. {@link InterestPlanRegistrationDTO} in input will be populated as below:  InterestPlanRegistrationDTO.interestRuleKeys: Map</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;String ,Date&gt; :- This is key value pair of the IRD and the effective date. Effective date here is the account opening date i.e. postingDate To get the list of IRDs linked to prouduct, call {@link IDDAProductAdapter#inquireProduct}</p> <p>InterestPlanRegistrationDTO.accountId = accountId (input) InterestPlanRegistrationDTO.depositNumber = ignore InterestPlanRegistrationDTO.partyId = partyId (input) InterestPlanRegistrationDTO.rate = ignore balanceFromDate = postingDate InterestPlanRegistrationDTO.balanceQualifier = ignore InterestPlanRegistrationDTO.balanceAmount = ignore InterestPlanRegistrationDTO.accountCurrency = accountCurrency InterestPlanRegistrationDTO.interestClass = ignore InterestPlanRegistrationDTO.interestDivision = ignore FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext domainCategory = DemandDeposit BankingFactContext.setProcessDate(Date processDate) BankingFactContext.setAccountCurrency(String accountCurrency) BankingFactContext.setAccountId(String accountId) BankingFactContext.setPartyId(String partyId) Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;; set InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param dDAInterestRegistrationDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param dDAInterestRegistrationDTO	<pre> This method contains the business logic for opening a demand deposit account based on the parameters <a href="#">{@link DemandDepositAccountOpeningDTO}</a> passed by the calling module[eg Origination].&lt;br&gt; &lt;ol&gt; The following is the algorithm for the method: &lt;li&gt;Perform all validations using <a href="#">{@link DemandDepositAccountBusinessPolicy#validate()}</a>. &lt;ul&gt;Following validations will be performed &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateProduct()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateProductExpiry()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePermissibleNoOfAccounts()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyId()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyRole()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateBranch()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateCurrency()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyBlacklisting()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateAccountRelationship()&amp;lt;/li&gt; &amp;lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateRestrictions()&amp;lt;/li&gt; &amp;lt;/ul&gt; &amp;lt;/li&gt;Generate the account number by using <a href="#">{@link DDAccountOpeningService#generateAccountNumber()&amp;lt;/li&gt; &amp;lt;li&gt;Call <a href="#">{@link this.deriveMinorAccountStatus()&amp;lt;/li&gt; to derive the minor status of the primary party and set </a></a></a></a></a></a></a></a></a></a></a></a></pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>netRateDerived : The information object carrying the details of the rate negotiated. See <a href="#">{@link NetRateDerived}</a> for more information.</p> <p><b>@param</b> partyToAccountRelationship[]: The information object carrying the party to account relationship.. See <a href="#">{@link PartyToAccountRelationship}</a> for more information.</p> <p><b>&lt;/ol&gt;</b> For Account Opening:Product level banking facilities cannot be overridden(If at product debit card flag is false, <b>&lt;br&gt;</b> at account level it cannot be set as true. If channelFacilitesDTO is null--&gt;banking facilities from product will be applicable. Maximum number of accounts that a party can have under the product as the primary (first) holder will be defined at the Product level. System should not permit CASA account opening if this account is breaching the set limit. System should permit CASA account opening only if the product code is active and not expired. It works as follows: <b>&lt;li&gt;</b>Get the count of account for which the party is primary account holder by calling <code>IDDAPartyAdapter#fetchCountPrimaryAccountHolder(partyId : String)</code> <b>&lt;/li&gt;</b> <b>&lt;li&gt;</b>Compare the count with the allowable count at the product level.<b>&lt;/li&gt;</b> <b>&lt;li&gt;</b>Reject account opening if the product level count is less than or equal to the count for which party is primary account holder. <b>&lt;/pre&gt;</b></p> <p><b>@param</b> demandDepositAccount <b>@param</b> interestTicketNumber <b>@param</b> partyToAccountRelationship <b>@param</b> isDocumentIncomplete <b>@param</b> isCheckingAccount <b>@throws</b> FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param dDAInterestRegistrationDTO	This method is used to perform different types of registration i.e registration with pricing, interest, NPA and taxation module @param demandDepositClonedAccount @param postingDate @param interestTicketNumber @param parentAccountId @throws FatalException
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param dDAInterestRegistrationDTO	<p>&lt;ol&gt;This method is responsible for registering interest plan in the interest module. This method will</p> <ul style="list-style-type: none"> <li>call the <code>{@link IDDAInterestAdapter#registerInterestPlan(accountId: String, partyId: String, accountCurrency: String, postingDate: Date): Map&lt;String, Date&gt;}</code></li> <li>logSettlementAction for logging the settlement action for the settlementDate using <code>{@link this.logSettlementAction(String accountId, String partyId, Map&lt;String, Date&gt;)}</code></li> </ul> <p>&lt;li&gt;&lt;b&gt;Since:&lt;/b&gt; R2.2 &lt;br&gt; a new attribute valueDate is introduced in DDAInterestRegistrationDTO which can be backdated in case of backdated account opening.&lt;/li&gt; &lt;/ol&gt;</p> <p>@param demandDepositAccount @param interestTicketNumber @param postingDate @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param dDAInterestRegistrationDTO	<p>This method contains the business logic for opening a backdated demand deposit account based on the parameters <a href="#">{@link DemandDepositAccountOpeningDTO}</a> passed by the calling module[eg Origination]. It contains the account opening date which is less than the current process date,it also contains the details of broker/introducer's which was used to open the backdated account.&lt;br&gt;&lt;ol&gt;The following is the algorithm for the method:&lt;li&gt;Fetch all the product level parameters for given offer id <a href="#">{@link OfferInquiryApplicationService#fetchCASAOffer()}&lt;/li&gt;&lt;/ol&gt;Perform all validations using <a href="#">{@link DemandDepositAccountBusinessPolicy#validate()}&lt;/li&gt;&lt;/ol&gt;Following validations will be performed in addition to the existing validations of account opening&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateBatchRunning()}&lt;/li&gt;&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePeriodForBackDatedAccount}&lt;/li&gt;&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateOfferForBackDated}&lt;/li&gt;&lt;/ol&gt; &lt;li&gt;Generate the account number by using <a href="#">{@link DDAccountOpeningService#generateAccountNumber()}&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link this.deriveMinorAccountStatus()}&lt;/li&gt; to derive the minor status of the primary party and set demandDepositAccount.AccountPreferences.minorAccountStatus = minorAccountStatus&lt;/li&gt; &lt;li&gt;AccountPreferences is a component in the DemandDepositAccount&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link this.setAccountStatus()}&lt;/li&gt; to initialize the SystemDefined and UserDefined account statuses&lt;/li&gt; &lt;li&gt;Call private method <a href="#">{@link</a></a></a></a></a></a></a></a></a></p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>setAgingStatusDates()} for initializing the aging statuses and dates.&lt;/li&gt; &lt;li&gt; Call {@link setDefaultAccountFacilities()} and {@link setDefaultAccountFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt; Call {@link setDefaultBankingFacilities()} and {@link setDefaultBankingFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt;Persist the account object to the database by calling {@link DemandDepositAccountRepository#insertDemandDepositAccount()}&lt;/li&gt; &lt;li&gt; Call private method {@link this.createPricingPlanAndEndDialogueForBackdate()} to create pricing plan in the pricing module using the back dated account opening date&lt;/li&gt; &lt;li&gt; Log periodic fee action for computation {@link Compute} for each of the dates returned above, by calling the service {@link PeriodicFeeLogService.logPeriodicFeeAction}. These actions will be picked up and processed as part of the periodic fee shell. Also in case the periodic fee charged is greater than backdated account opening date then log action.&lt;/li&gt; &lt;li&gt; Call private method {@link this.registerInterestPlanWithBackdate()} to registering interest plan in the interest module with a backdated date(takes value date).&lt;/li&gt; &lt;li&gt; Call private method {@link this.registerForTaxationBackdated()} to register the newly opened account in the Tax module which should now also take account opening date as input.&lt;/li&gt; &lt;li&gt; Call private method {@link this.notifyNPAForBackdatedAccount()} to register the newly opened backdated account in the NPA module which should now also take account opening date as input.&lt;/li&gt;The following are the parameters taken by the method along with their significance&lt;ul&gt; &lt;li&gt; Call private method {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>MinimumAverageBalanceHelper#initializeBackdatedAverageBalances(DemandDepositAccount, Date)} to initialize the average balances using the backdated account opening date.&lt;/li&gt; &lt;li&gt; Call {@link EventRateApplicationService#persistRates()} through DDAPricingAdapter to fetch and stamp the net rates for casa interest stages and call {@link WrappedEventRateApplicationServiceLog#getLenientRatesByPlanIdWithApplicableUDM()} to derive UDM's and Margin rates and log rate change action.&lt;li&gt;Call updateBundleSubscription of the BundleAccountLinkageService to intimate the bundle subscription unit to update bundle status and benefits, when an account is being opened&lt;/li&gt;&lt;li&gt; call updateCollectionForAccount to intimate collections about the accrual status of the account.&lt;/li&gt;&lt;/ul&gt;&lt;/ol&gt;&lt;/pre&gt;</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer {@link IDDAInterestAdapter#registerInterestPlan} @param accountId @param partyId @param accountCurrency @param postingDate	<p>The method is responsible for registering an interest plan for the account in the interest module at the time of account opening. This method is referred from {@link DDAccountOpeningService#openDemandDepositAccount()} Call {@link registerInterestPlan(ctx: SessionContext, interestRegistrationDTO: InterestPlanRegistrationDTO, factContext: FactContext): InterestPlanRegistrationResponse } to register in interest module. InterestPlanRegistrationResponse contains a map of settlementDate for each IRD linked to the product. {@link InterestPlanRegistrationDTO} in input will be populated as below: InterestPlanRegistrationDTO.interestRuleKeys: Map &lt;String ,Date&gt; :- This is key value pair of the IRD and the effective date. Effective date here is the account</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>opening date i.e. postingDate To get the list of IRDs linked to product, call {@link IDDAProductAdapter#inquireProduct}</p> <p>InterestPlanRegistrationDTO.accountId = accountId  (input) InterestPlanRegistrationDTO.depositNumber = ignore  InterestPlanRegistrationDTO.partyId = partyId  (input) InterestPlanRegistrationDTO.rate = ignore  balanceFromDate = postingDate  InterestPlanRegistrationDTO.balanceQualifier = ignore  InterestPlanRegistrationDTO.balanceAmount = ignore  InterestPlanRegistrationDTO.accountCurrency = accountCurrency  InterestPlanRegistrationDTO.interestClass = ignore  InterestPlanRegistrationDTO.interestDivision = ignore  FactContext will be populated as below: create an object of BankingFactContext(domainCategory) which extends FactContext domainCategory = DemandDeposit  BankingFactContext.setProcessDate(Date processDate)  BankingFactContext.setAccountCurrency(String accountCurrency)  BankingFactContext.setAccountId(String accountId)  BankingFactContext.setPartyId(String partyId) Return type InterestPlanRegistrationResponse contains array of InterestPlanRegistrationResponseDTO. Iterate through the array and populate Map&lt;String, Date&gt;;  set  InterestPlanRegistrationResponseDTO.interestClass in the first argument of the map and set  InterestPlanRegistrationResponseDTO.settlementDate in the second argument of the map. @param dDAInterestRegistrationDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param accountId @param partyId @param accountCurrency @param postingDate	<pre> This method contains the business logic for opening a demand deposit account based on the parameters <a href="#">{@link DemandDepositAccountOpeningDTO}</a> passed by the calling module[eg Origination].&lt;br&gt; &lt;ol&gt; &lt;li&gt;Perform all validations using <a href="#">{@link DemandDepositAccountBusinessPolicy#validate()}</a>. &lt;ul&gt; &lt;li&gt;Following validations will be performed &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateProduct()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateProductExpiry()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePermissibleNoOfAccounts()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyId()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyRole()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateBranch()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateCurrency()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePartyBlacklisting()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateAccountRelationship()}</a> &lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateRestrictions()}</a> &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; Generate the account number by using <a href="#">{@link DDAccountOpeningService#generateAccountNumber()}</a>. &lt;li&gt;Call <a href="#">{@link this.deriveMinorAccountStatus()}</a> to derive the minor status of the primary party and set </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>netRateDerived : The information object carrying the details of the rate negotiated. See <a href="#">{@link NetRateDerived}</a> for more information.</p> <p><b>@param</b> partyToAccountRelationship[]: The information object carrying the party to account relationship.. See <a href="#">{@link PartyToAccountRelationship}</a> for more information.</p> <p><b>&lt;/ol&gt;</b> For Account Opening:Product level banking facilities cannot be overridden(If at product debit card flag is false, <b>&lt;br&gt;</b> at account level it cannot be set as true. If channelFacilitesDTO is null--&gt;banking facilities from product will be applicable. Maximum number of accounts that a party can have under the product as the primary (first) holder will be defined at the Product level. System should not permit CASA account opening if this account is breaching the set limit. System should permit CASA account opening only if the product code is active and not expired. It works as follows: <b>&lt;li&gt;</b>Get the count of account for which the party is primary account holder by calling <code>IDDAPartyAdapter#fetchCountPrimaryAccountHolder(partyId : String)</code><b>&lt;/li&gt;</b> <b>&lt;li&gt;</b>Compare the count with the allowable count at the product level.<b>&lt;/li&gt;</b> <b>&lt;li&gt;</b>Reject account opening if the product level count is less than or equal to the count for which party is primary account holder. <b>&lt;/pre&gt;</b></p> <p><b>@param</b> demandDepositAccount <b>@param</b> interestTicketNumber <b>@param</b> partyToAccountRelationship <b>@param</b> isDocumentIncomplete <b>@param</b> isCheckingAccount <b>@throws</b> FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param accountId @param partyId @param accountCurrency @param postingDate	This method is used to perform different types of registration i.e registration with pricing, interest, NPA and taxation module @param demandDepositClonedAccount @param postingDate @param interestTicketNumber @param parentAccountId @throws FatalException
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param accountId @param partyId @param accountCurrency @param postingDate	<p>&lt;ol&gt;This method is responsible for registering interest plan in the interest module. This method will</p> <ul style="list-style-type: none"> <li>call the <code>{@IDDAInterestAdapter#registerInterestPlan(accountId: String, partyId: String, accountCurrency: String, postingDate: Date): Map&lt;String, Date&gt;}</code></li> <li>logSettlementAction for logging the settlement action for the settlementDate using <a href="#">{@link this.logSettlementAction(String accountId, String partyId, Map&lt;String, Date&gt;)}</a></li> <li><b>Since:</b> R2.2 a new attribute valueDate is introduced in IDDAInterestRegistrationDTO which can be backdated in case of backdated account opening.</li> </ul> <p>@param demandDepositAccount @param interestTicketNumber @param postingDate @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	registerInterestPlan	Refer <a href="#">{@link IDDAInterestAdapter#registerInterestPlan}</a> @param accountId @param partyId @param accountCurrency @param postingDate	This method contains the business logic for opening a backdated demand deposit account based on the parameters <a href="#">{@link DemandDepositAccountOpeningDTO}</a> passed by the calling module[eg Origination]. It contains the account opening date which is less than the current process date,it also contains the details of broker/introducer's which was used to open the backdated account. <ol>The following is the algorithm for the method:<li>Fetch all the product level parameters for given offer id <a href="#">{@link OfferInquiryApplicationService#fetchCASAOffer()}&lt;/li&gt;&lt;/ol&gt;Perform all validations using <a href="#">{@link DemandDepositAccountBusinessPolicy#validate()}&lt;/li&gt;&lt;/ul&gt;Following validations will be performed in addition to the existing validations of account opening&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateBatchRunning()}&lt;/li&gt;&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validatePeriodForBackDatedAccount}&lt;/li&gt;&lt;li&gt;<a href="#">{@link DemandDepositAccountBusinessPolicy#validateOfferForBackDated}&lt;/li&gt;&lt;/ul&gt; &lt;li&gt;Generate the account number by using <a href="#">{@link DDAccountOpeningService#generateAccountNumber()}&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link this.deriveMinorAccountStatus()}&lt;/li&gt; to derive the minor status of the primary party and set demandDepositAccount.AccountPreferences.minorAccountStatus = minorAccountStatus&lt;/li&gt; &lt;li&gt;AccountPreferences is a component in the DemandDepositAccount&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link this.setAccountStatus()}&lt;/li&gt; to initialize the SystemDefined and UserDefined account statuses&lt;/li&gt; &lt;li&gt;Call private method <a href="#">{@link</a></a></a></a></a></a></a></a></a>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>setAgingStatusDates() for initializing the aging statuses and dates.</p> <ul style="list-style-type: none"> <li>Call <a href="#">{@link setDefaultAccountFacilities()}</a> and <a href="#">{@link setDefaultAccountFacilities()}</a> for setting up the default values</li> <li>Call <a href="#">{@link setDefaultBankingFacilities()}</a> and <a href="#">{@link setDefaultBankingFacilities()}</a> for setting up the default values</li> <li>Persist the account object to the database by calling <a href="#">{@link DemandDepositAccountRepository#insertDemandDepositAccount()}</a></li> <li>Call private method <a href="#">{@link this.createPricingPlanAndEndDialogueForBackdate()}</a> to create pricing plan in the pricing module using the back dated account opening date</li> <li>Log periodic fee action for computation <a href="#">{@link Compute}</a> for each of the dates returned above, by calling the service <a href="#">{@link PeriodicFeeLogService.logPeriodicFeeAction}</a>. These actions will be picked up and processed as part of the periodic fee shell. Also in case the periodic fee charged is greater than backdated account opening date then log action.</li> <li>Call private method <a href="#">{@link this.registerInterestPlanWithBackdate()}</a> to registering interest plan in the interest module with a backdated date(takes value date).</li> <li>Call private method <a href="#">{@link this.registerForTaxationBackdated()}</a> to register the newly opened account in the Tax module which should now also take account opening date as input.</li> <li>Call private method <a href="#">{@link this.notifyNPAForBackdatedAccount()}</a> to register the newly opened backdated account in the NPA module which should now also take account opening date as input.</li> </ul> <p>The following are the parameters taken by the method along with their significance</p> <ul style="list-style-type: none"> <li>Call private method <a href="#">{@link</a></li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>MinimumAverageBalanceHelper#initializeBackdatedAverageBalances(DemandDepositAccount, Date)} to initialize the average balances using the backdated account opening date.&lt;/li&gt; &lt;li&gt; Call {@link EventRateApplicationService#persistRates()} through DDAPricingAdapter to fetch and stamp the net rates for casa interest stages and call {@link WrappedEventRateApplicationServiceLog#getLenientRatesByPlanIdWithApplicableUDM()} to derive UDM's and Margin rates and log rate change action.&lt;/li&gt;Call updateBundleSubscription of the BundleAccountLinkageService to intimate the bundle subscription unit to update bundle status and benefits, when an account is being opened&lt;/li&gt;&lt;li&gt; call updateCollectionForAccount to intimate collections about the accrual status of the account.&lt;/li&gt;&lt;/ul&gt;&lt;/ol&gt;&lt;/pre&gt;</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doBackDateInterestProcessing	This service is used for recomputation of interest and posting of adjustment entries for an account if backdated transaction occurs on an account. @param applicationContext @param backDatedInterestProcessingRequestDTO	This service is used for recomputation of interest and posting of adjustment entries for an account if backdated transaction occurs on an account. For example, DD module performs backdated cash deposit transaction.  Due to this the base balance for interest computation would change for all subsequent interest computations happened after value date of backdated transaction. Hence interest needs to be recomputed for all subsequent balance changes which happened after value date of backdated transaction.  Also if backdated transaction has crossed current Accrual/Compounding/Capitalization/Settlement cycles then all these interest phases needs to be reprocessed and Interest adjustment entries needs to be passed for differential interest amount.  This method works as follows: <li>Convert the input DTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>to balance record entity by invoking  {@BalanceRecordDTO.convertDTOToEntity() }&lt;/li&gt;  &lt;li&gt;Create instance of BalanceChangeAnalyser using the input.applicationContext&lt;/li&gt; &lt;li&gt;Create the factContext with factMap, DDAAccount.AccountId as parameter. Retrieve the account Id from the input backDatedInterestProcessingRequestDTO&lt;/li&gt;  &lt;li&gt;Construct BackDatedInterestProcessingRequest with the values fetched or retrieved above&lt;/li&gt;  &lt;li&gt;Invoke the service {@link BackDatedInterestProcessingService#doBackDatedInterestProcessing} to preform back dated interest processing.&lt;/li&gt; &lt;li&gt;Validate backDatedInterestProcessingResponse. If the status code is not success, throw Business Exception.&lt;/li&gt;  &lt;li&gt;Get InterestPlanItemProcessingResult from BackDatedInterestProcessingResponse returned by the above service&lt;/li&gt; &lt;li&gt;Retrieve the InterestPlanItemKey and BackDatedInterestSettlementResponse from the result and populate InterestPlanItemKeyDTO,List&lt;BackDatedInterestSettlementResponseDTO&gt; map. &lt;li&gt; &lt;li&gt; Return back Dated Interest Settlement Response Map&lt;/li&gt;  @param applicationContext @param backDatedInterestProcessingRequestDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doBackdatedInterestProcessing	This service is used for recomputation of interest and posting of adjustment entries for an account if backdated transaction occurs on an account. @param applicationContext @param backDatedInterestProcessingRequestDTO	This method performs actual backdated processing for interest computation. It is created so as to facilitate internal recursive calls to this method.  When a backdated transaction takes place on an account, interest has to be recomputed for the account.  In case, the transaction value date is beyond the settlement date, then adjustment entries have to be passed for the settlement date and recomputation has to be done with the updated balances.   In order to facilitate recomputation, the caller module (CASA in this case) , prepares a set of balance records along with the interest rates from the value date of the transaction upto the posting date.   This set of records is sent to the interest module to enable it to perform computations. Along with the balance records, an instance of the BalancechangeAnalyser is also sent. This class, which implements the interface IBalanceChangeAnalyser, consists of a service to settle the adjustment amount and prepare the balance records with the updated balances of the account.  In effect, the interest service {@link IDDAInterestAdapter#doBackdatedInterestProcessing} recomputes the interest from the value date, settles the adjustment amount and recomputes interest again beyond the settlement date. It is a recursive process and continues till the current processing date.   As part of the alternate account maintenance, if an alternate account is maintained for InterestPosting of the input.account, then the adjustment entry has to be posted to the alternate account.   This will result in a backdated transaction of the alternate account, and hence recomputation of interest of the alternate account has to be triggered. It might so happen that the entry results in a beyond cap transaction for the alternate account. Hence, the whole process of

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>recomputation explained above has to be performed for the alternate account as well. &lt;br&gt; Therefore, this method has been designed in a fashion to support recursive calls for interest recomputation. &lt;br&gt;The method works as follows: &lt;li&gt;Call the method <a href="#">{@link BackdatedProcessingHelper#populateBalanceRecords}</a> to populate the balance records as required by the interest service.&lt;/li&gt; &lt;li&gt;Prepare the <a href="#">BackdatedInterestProcessingRequestDTO</a> with the balanceRecords returned by the service above, account id and requested date. The instance of the class <a href="#">{@link DDABalanceChangeAnalyser}</a>, which is required by the back dated interest processing request is later set in the adapter impl class.&lt;/li&gt; &lt;li&gt;Call the service <a href="#">{@link IDDAInterestAdapter#doBackdatedInterestProcessing}</a> with the request created in the step above to recalculate and process the backdated interest.&lt;/li&gt; &lt;li&gt;Validate the response returned by the interest service. Open a loop for the <a href="#">BackDatedInterestSettlementResponseDTO</a> response. &lt;ol&gt; &lt;li&gt;If the(<a href="#">InterestClass = Credit</a>, <a href="#">AlternateAccount#AccountTypeForCreditPosting = DemandDeposit</a> and <a href="#">AlternateAccount#AccountIdForCreditPosting</a> &lt;&gt; null) OR (ElsIf the <a href="#">interestClass = Debit</a>, <a href="#">AlternateAccount#AccountTypeForDebitPosting = DemandDeposit</a> and <a href="#">AlternateAccount#AccountIdForDebitPosting</a> &lt;&gt; null), then &lt;li&gt;Open a loop for the <a href="#">BackDatedInterestSettlementResponseDTO</a> &lt;li&gt;If the value of <a href="#">(BackDatedInterestSettlementResponse#oldPostTaxInterestAmount</a> &lt;&gt; <a href="#">BackDatedInterestSettlementResponse#newPostTaxInt</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>erestAmount), it means that an adjustment entry has been passed for the settlement date. Since the adjustment entry has hit the alternate account, backdated interest computation for the alternate account also needs to be triggered. &lt;li&gt;Call the service {<a href="#">@link</a> BackdatedTransactionExecutor#performBackdatedInterestProcessing} with the effectiveDate as the BackDatedInterestSettlementResponse#dueDate and the accountId as the AccountIdForCreditPosting OR AccountIdForDebitPosting as the case may be.&lt;/li&gt;&lt;li&gt;Exit loop for BackdatedSettlementResponseDTO. The records in the BackdatedSettlementResponse will be in ascending order of dueDate. So, the interest needs to be recomputed from the earliest date on which an adjustment has been made. And hence, the call to the service is required only once.&lt;/li&gt;&lt;/li&gt;&lt;/ol&gt;&lt;/li&gt; @param applicationContext @param effectiveDate @param accountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	inquireInterestAtArrangementClosure	This service is used to inquire the interest details at the time of account closure. It takes into consideration the IRD level parameters such as <a href="#">{@link ComputationTers#excludeAccountClosingDate}</a> , <a href="#">{@link ComputationTers#excludeAccountClosingMonth}</a> and computes the interest accordingly. In turn it calls the interest service <a href="#">{@link ArrangementClosureApplicationService#inquireInterestAtArrangementClosure}</a> to do the processing. @param sessionContext @param arrangementNumber @param productClass @param interestClass @param interestDivision @return @throws FatalException	This service is used to inquire the interest details at the time of account closure. It takes into consideration the IRD level parameters such as <a href="#">{@link ComputationTers#excludeAccountClosingDate}</a> , <a href="#">{@link ComputationTers#excludeAccountClosingMonth}</a> and computes the interest accordingly. In turn it calls the interest service <a href="#">{@link ArrangementClosureService#inquireInterestAtArrangementClosure}</a> to do the processing. @param sessionContext @param arrangementNumber @param productClass @param interestClass @param interestDivision @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	inquireInterestAtArrangementClosure	This service is used to inquire the interest details at the time of account closure. It takes into consideration the IRD level parameters such as {@link ComputationTers#excludeAccountClosingDate}, {@link ComputationTers#excludeAccountClosingMonth} and computes the interest accordingly. In turn it calls the interest service {@link ArrangementClosureApplicationService#inquireInterestAtArrangementClosure} to do the processing. @param sessionContext @param arrangementNumber @param productClass @param interestClass @param interestDivision @return @throws FatalException	This method prepares the response using the Account Balances available for the provided Account. @param transactionDates @param demandDepositAccount @param accountBalancesMap @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	updateSystemDefineStatus	Method will be used to update SystemDefineStatus of DemandDepositAccount in interest module.  Whenever there is a System Define status change on DDAccount ,this method will be called to inform Interest about the same. @param accountId @param accountStatus	This method is used to modify {@link SystemDefinedStatus} of the demand deposit account.This method also does auditing activity. This method will perform activity is following sequence. <ol> <li>oldSystemDefinedStatus = this.systemDefinedStatus. <li>if oldSystemDefinedStatus equal to input.systemDefinedStatus then return <li>this.systemDefinedStatus = input.systemDefinedStatus. <li>Create {@link SystemDefinedStatusAuditItem} with following specification <ol> <li>accountId = this.key.accountId <li>fromEffectiveDate = oldSystemDefinedStatus.effectiveDate <li>toEffectiveDate = input.effective date. <li>fromProcessDate = oldSystemDefinedStatus.processDate <li>toProcessDate = input.processDate. <li>systemDefinedStatus = oldSystemDefinedStatus <li>user number = get it from thread local->Session context. </ol> @param newStatus @param effectiveDate @param isInterestIntimationRequired
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	updateSystemDefineStatus	Method will be used to update SystemDefineStatus of DemandDepositAccount in interest module.  Whenever there is a System Define status change on DDAccount ,this method will be called to inform Interest about the same. @param accountId @param accountStatus	Method will be used to update SystemDefineStatus of DemandDepositAccount in interest module.  Whenever there is a System Define status change on DDAccount ,this method will be called to inform Interest about the same. @param accountId @param accountStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	updateSystemDefineStatus	Method will be used to update SystemDefineStatus of DemandDepositAccount in interest module.  Whenever there is a System Define status change on DDAccount ,this method will be called to inform Interest about the same. @param accountId @param accountStatus	When account status changes from <b> Dormant </b> to <b> Unclaimed </b>, credit and debit Settlement action takes place.   Following are the key points. <ol> <li>This is adhoc settlement similar to account closure. <li>Credit as well as Debit settlement(if applicable) will be performed. <li>Interest will be computed till process date. <li>If account is in debit balance , account will not be marked as unclaimed. Idea is to recover all the funds before marking status as unclaimed. <li>Once account is unclaimed, we can not perform any transaction. </ol> @param applicationContext @param demandDepositAccount
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	intimateInterestForAccountTransfer	Method for Account Branch Transfer @reviseInterestForBrokenPeriod @param balanceRecordsSet @return	Method intimateInterest will be used to intimate the interests while transferring the account from one branch to other branch. @param accountId @param casaDepositNumber @param newBranchId @param transferDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getInterestPeriodForCurrentAndNextCycle	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @param interestClass @return interestCyclePeriodResponse	This method will be used to execute bonus interest processing.  Sequence for execution-  <ol><li>Inquire product id by loading DDAccount.<li>check for UDM maintenance at product level if present then <li>Get all the facts from consulting to evaluate the rule. <li>Validate rule for bonus interest eligibility and if account is eligible for bonus interest then <li>log the non financial statement item with description. <li>check if bonus interest is applicable for current cap cycle then call Interest module to get the start date and end date of current cycle and Call pricing to create back dated UDM Definition at account level, provide start date and end date also.  and instead of logging backdated action process backdated right away. <li>And if bonus interest is applicable for next cap cycle then create future dated UDM with start and end date and log the rate change action with due date as start date. </ol> @param applicationContext Application Context which will be set in batch process. @param actionKey @param actionRequest every subclass need to cast this to appropriate subclass of {@link ActionKey} @param dueDate due date for the action
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getInterestPeriodForCurrentAndNextCycle	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @param interestClass @return interestCyclePeriodResponse	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @return interestCyclePeriodResponse
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getCurrentCapitalizationStatistics		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getCurrentCapitalizationStatistics		This method fetches interest parameter details by calling interest module and sets it to the DebitInterestParametersDetailsDTO and CreditInterestParametersDetailsDTO.
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getRevisedCapitalizationStatistics		
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	getRevisedCapitalizationStatistics		This method gives Revised Capitalisation Dates on change of capitalisation frequency, anniversary day and effective cycle.
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	modifyCapitalisationStatistics		Debit interest capitalisation statistics stamped during account opening can be modified at account level. This method calls method modifyCapitalisationStatistics from service interestOperationService through ddaInterestAdapter to modify the debit capitalisation statistics.
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	modifyCapitalisationStatistics		aCredit interest capitalisation statistics stamped during account opening can be modified at account level. This method calls method modifyCapitalisationStatistics from service interestOperationService through ddaInterestAdapter to modify the credit capitalisation statistic.
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	inquireAccrualInterestDetails		This will provide the Interest on the Account which is Accrued interest. @param accountId @param accountArrearResponseType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	stopSuspendedAccrual	<p>For the given arrangement number provided in the input, the accrual status of all the interest plan item registered with that account number will be marked as suspended.</p> <p>&lt;br&gt;This service will internally call  {@link InterestOperationService#stopSuspendedAccrual(String, InterestClass, InterestDivision, Date)} &lt;ol&gt;  &lt;li&gt;Load the {@link InterestPlan}  &lt;li&gt;Fetch the interest plan items from the interest plan &lt;li&gt;Loop through the interest plan items  &lt;li&gt;Call {@link InterestOperationService#stopSuspendedAccrual(String, InterestClass, InterestDivision, Date, Date, String)}  &lt;/ol&gt; @param accountId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	stopSuspendedAccrual	<p>For the given arrangement number provided in the input, the accrual status of all the interest plan item registered with that account number will be marked as suspended.</p> <p>&lt;br&gt;This service will internally call {<a href="#">@link InterestOperationService#stopSuspendedAccrual(String, InterestClass, InterestDivision, Date)</a>} &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Load the {<a href="#">@link InterestPlan</a>}</li> <li>&lt;li&gt;Fetch the interest plan items from the interest plan</li> <li>&lt;li&gt;Loop through the interest plan items</li> <li>&lt;li&gt;Call {<a href="#">@link InterestOperationService#stopSuspendedAccrual(String, InterestClass, InterestDivision, Date, Date, String)</a>}</li> </ul> <p>&lt;/ol&gt; @param accountId</p>	<p>This updates account as noranal. @param applicationContext @param accountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	startSuspendedAccrual	<p>For the given arrangement number provided in the input, the accrual status of all the interest plan item registered with that account number will be marked as suspended.</p> <p>&lt;br&gt;This service will internally call {<a href="#">@link InterestOperationService#startSuspendedAccrual(String, InterestClass, InterestDivision, Date, boolean)</a>}</p> <p>&lt;ol&gt; &lt;li&gt;Load the {<a href="#">@link InterestPlan</a>} &lt;li&gt;Fetch the interest plan items from the interest plan &lt;li&gt;Loop through the interest plan items &lt;li&gt;Call {<a href="#">@link InterestOperationService#startSuspendedAccrual(String, InterestClass, InterestDivision, Date, boolean)</a>}</p> <p>&lt;li&gt;Return the response to the caller &lt;/ol&gt; @param accountId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	startSuspendedAccrual	<p>For the given arrangement number provided in the input, the accrual status of all the interest plan item registered with that account number will be marked as suspended.</p> <p>&lt;br&gt;This service will internally call  {@link InterestOperationService#startSuspendedAccrual(String, InterestClass, InterestDivision, Date, boolean)}</p> <p>&lt;ol&gt; &lt;li&gt;Load the {@link InterestPlan} &lt;li&gt;Fetch the interest plan items from the interest plan &lt;li&gt;Loop through the interest plan items &lt;li&gt;Call {@link InterestOperationService#startSuspendedAccrual(String, InterestClass, InterestDivision, Date, boolean)} &lt;li&gt;Return the response to the caller &lt;/ol&gt; @param accountId</p>	<p>This marks account as suspended. @param applicationContext @param accountId @param arrearDTO</p>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDDAInterestCapitalization	<p>This method will be used to trigger the interest capitalisation for DDAccount. @param applicationContext @param accountId @param dueDate @param interestClass @param interestDivision @param effectiveDate @param isAccountClosed @return interestCapitalizationDetailsDTO</p>	<p>This method will be used to trigger the credit interest capitalization. @param applicationContext @param accountId @param settlementDate @param isAccountClosed @return InterestCapitalizationDetailsDTO</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDDAInterestCapitalization	This method will be used to trigger the interest capitalisation for DDAccount. @param applicationContext @param accountId @param dueDate @param interestClass @param interestDivision @param effectiveDate @param isAccountClosed @return interestCapitalizationDetailsDTO	This method will be used to trigger the debit interest capitalization and will raise the arrear of type {@link ArrearType#INTEREST_CAPITALISED_NOT_DUE}. @param applicationContext @param accountId @param settlementDate @param isAccountClosed @return InterestCapitalizationDetailsDTO
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	doDDAInterestCapitalization	This method will be used to trigger the interest capitalisation for DDAccount. @param applicationContext @param accountId @param dueDate @param interestClass @param interestDivision @param effectiveDate @param isAccountClosed @return interestCapitalizationDetailsDTO	This method will be used to trigger the interest capitalisation for DDAccount. @param applicationContext @param accountId @param dueDate @param interestClass @param interestDivision @param effectiveDate @param isAccountClosed @return interestCapitalizationDetailsDTO
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	freezeInterestHardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is <code>FreezeInterest</code> and interest rate type is "fixed" then it will call Interest module for HardShip Relief action for freezing interest. It will call the InterestFreezeService to freeze the interest if overdraft account has fixed rate. @param accountId @param reliefStartDate @param reliefExpiryDate @param reliefProcessDate @throws FatalException	This method will be called from hardShipReliefExecution if hardShipReliefType is "Freeze Interest" and interest rate type is "fixed" then it will call pricing adapter for HardShip Relief action for freezing interest. It will call the InterestFreezeService to freeze the interest if overdraft account has fixed rate. @param applicationContext @param hardShipReliefDetails @param updationType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	cancelHardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is "Freeze Interest" and HardshipReliefUpdataionType is "Extension" then it will call Interest module for HardShip Relief action for canceling freez interest. It will call the InterestFreezeService to cancel the freeze interest if overdraft account has fixed rate. @param accountId @param reliefExpiryDate @param reliefProcessDate @param isCallForModification @throws FatalException	This method will be called from hardShipReliefExecution if hardShipReliefType is "Freeze Interest" and interest rate type is "fixed" then it will call pricing adapter for HardShip Relief action for freezing interest. It will call the InterestFreezeService to freeze the interest if overdraft account has fixed rate. @param applicationContext @param hardShipReliefDetails @param updationType
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	interestBenefitInquiryResponse	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @param interestClass @return interestCyclePeriodResponse	This method will be used to inquire about the interest benefit period @param accountId @param effectiveDate @return interestCyclePeriodResponse
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	interestBenefitInquiryResponse	This method will be used to inquire about the interest current cycle period and the next cycle period. @param accountId @param effectiveDate @param interestClass @return interestCyclePeriodResponse	This method will be used to log interest benefit action on account having offset facility.  This action will be executed in batch and will be used to calculate the interest benefit enjoyed by   account due to having offset facility. @param demandDepositAccount
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	writeOffAccount	@param appContext @param accountId @return	This method calls the processWriteOff() method of InterestWriteOffService. It should be called during full writeoff of the account. @param appContext @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc												
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	writeOffAccount	@param appContext @param accountId @return	<p>This service do respective accounting for account write of</p> <table border="1"> <thead> <tr> <th data-bbox="1295 326 1493 358">Event Code</th> <th data-bbox="1493 326 1703 358">Dr Transaction Code</th> <th data-bbox="1703 326 1946 358">Account Role Code - Debit</th> <th data-bbox="1946 326 1959 358"></th> </tr> </thead> <tbody> <tr> <td data-bbox="1295 358 1493 391">Netting Indicator</td> <td data-bbox="1493 358 1703 391">Account Role Code - Debit</td> <td data-bbox="1703 358 1946 391">Debit GL Status</td> <td data-bbox="1946 358 1959 391"></td> </tr> <tr> <td data-bbox="1295 391 1493 423">Amount Tag - Debit</td> <td data-bbox="1493 391 1703 423">Cr Transaction Code</td> <td data-bbox="1703 391 1946 423"></td> <td data-bbox="1946 391 1959 423"></td> </tr> </tbody> </table>	Event Code	Dr Transaction Code	Account Role Code - Debit		Netting Indicator	Account Role Code - Debit	Debit GL Status		Amount Tag - Debit	Cr Transaction Code		
Event Code	Dr Transaction Code	Account Role Code - Debit														
Netting Indicator	Account Role Code - Debit	Debit GL Status														
Amount Tag - Debit	Cr Transaction Code															

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> align="CENTER"&gt;&lt;b&gt; &lt;span style="font-family: Arial; font-size: 12pt; color: rgb(255, 255, 255);"&gt;Account Role Code - Credit&lt;/span&gt;&lt;/b&gt;&lt;/td&gt; &lt;td width="143" bgcolor="#808080" align="CENTER"&gt;&lt;b&gt; &lt;span style="font-family: Arial; font-size: 12pt; color: rgb(255, 255, 255);"&gt;Credit GL Status&lt;/span&gt;&lt;/b&gt;&lt;/td&gt; &lt;td width="318" bgcolor="#808080" align="CENTER"&gt;&lt;b&gt; &lt;span style="font-family: Arial; font-size: 12pt; color: rgb(255, 255, 255);"&gt;Amount Tag - Credit&lt;/span&gt;&lt;/b&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P45&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_PROVISION_LIAB&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NA&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_PROVISION_LIAB_AMT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P46&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_PROVISION_LIAB_AMT&lt;/span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P49&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_WRITEOFF_EXPENSE&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NA&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NET_WRITEOFF_AMT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P50&lt;/span&gt;&lt;/td&gt; &lt;td </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NET_WRITEOFF_AMT&lt;/span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="TOP" height="64" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;R9&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NRML_REG_INT_CAP_WRITEOFF_AMT&lt;/span&gt;&lt; /td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;R10&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NRML_REG_INT_CAP_WRITEOFF_AMT&lt;/span&gt;&lt; /td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="TOP" height="64" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;R11&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSP_REG_INT_CAP_WRITEOFF_AMT&lt;/span&gt;&lt;/ td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;R12&lt;/span&gt;&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSP_REG_INT_CAP_WRITEOFF_AMT&lt;/span&gt;&lt;/ td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P53&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_WRITEOFF_EXPENSE&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NA&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NRML_REG_INT_CHRG_WRITEOFF_AMT &lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P54&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_RECEIVABLE&lt;/span&gt;&lt; /td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NRML_REG_INT_CHRG_WRITEOFF_AMT &lt;/span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P55&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_SUSP_REG_INT_CHRG_WRITEOFF_AMT&lt; /td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P56&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_RECEIVABLE&lt;/span&gt;&lt; /td&gt; &lt;td align="LEFT"&gt;&lt;span </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_SUSP_REG_INT_CHRG_WRITEOFF_AMT&lt; /span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P57&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_WRITEOFF_EXPENSE&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NRML_REG_INT_ACCR_WRITEOFF_AMT &lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P58&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_ACCRUED&lt;/span&gt;&lt;/t d&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_NRML_REG_INT_ACCR_WRITEOFF_AMT &lt;/span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;P59&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_SUSP_REG_INT_ACCR_WRITEOFF_AMT&lt; /span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;P60&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;CASA_DR_REGULARINT_ACCRUED&lt;/span&gt;&lt;/t d&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> align="LEFT"&gt;&lt;span "&gt;CASA_SUSP_REG_INT_ACCR_WRITEOFF_AMT&lt; /span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="TOP" height="64" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;R13&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASA_ECOCOST_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSP_ECOCOST_WRITEOFF_AMT_SUSP&lt;/span&gt;&lt; /td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;R14&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSP_ECOCOST_WRITEOFF_AMT_SUSP&lt;/span&gt;&lt; /td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="TOP" height="64" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;R13&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASA_ECOCOST_INCOME&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;SUSP_ECOCOST_WRITEOFF_AMT_NRML&lt;/span&gt; &lt;/td&gt; &lt;td align="LEFT"&gt;&lt;span "&gt;R14&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;span "&gt;ACCOUNT&lt;/span&gt;&lt;/td&gt; </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td align="LEFT"&gt;&lt;span &gt;SUSP_ECOCOST_WRITEOFF_AMT_NRML&lt;/span&gt; &lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="BOTTOM" height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;xx&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;PDID_INCOME &lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;SUSPENDED&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;PDID_WRITEOFF_SUSPAMT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;br /&gt; &lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;CASAACCOUNT&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;Account&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;PDID_WRITEOFF_SUSPAMT&lt;/span&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td valign="BOTTOM" height="19" align="LEFT"&gt;&lt;span "&gt;DD_ACCOUNT_WRITEOFF&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;xx&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;N&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;PDID_INCOME &lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;NORMAL&lt;/span&gt;&lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span "&gt;PDID_WRITEOFF_AMT&lt;/span&gt;&lt;/td&gt; &lt;td valign="TOP" align="LEFT"&gt;&lt;br /&gt; &lt;/td&gt; &lt;td valign="BOTTOM" align="LEFT"&gt;&lt;span </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				">CASAACCOUNT</span></td> <td valign="BOTTOM" align="LEFT"><span ">Account</span></td> <td valign="BOTTOM" align="LEFT"><span ">PDID_WRITEOFF_AMT</span></td> </tr> </tbody> </table>   @param applicationContext @param accountId @param provisionDrawdownAmount @param isFullWriteOff TODO @param arrearWriteOffHistoryResponce @return
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	checkIfTheAccountIsInExcludedStatusInIRD	To check whether the account is in excluded status in IRD	This is the public method exposed in the class. It internally calls the private methods to perform theValidations. The private methods called are : <ol><li>validateAccountLinkages</li><li>validateAccountFacilities</li><li>validateAccountBalances</li><li>validateStatus</li> </ol>
IDDAInterest Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAInterestAdapter	checkIfTheAccountIsInExcludedStatusInIRD	To check whether the account is in excluded status in IRD	to check whether the account status is in excluded status in IRD.
IBranchOpsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bropp\IBranchOpsAdapter	getEodBodCutOffStatus	<p>This method returns the status for EOD, BOD and Cut-Off indicating whether each one completed successfully or not. @param sessionContext the session context object. @param date the date for which the status is required. @return the <tt>EodBodCutOffStatusResponse</tt> object. @throws FatalException	<p> Returns whether the branch opening is allowed for the given date. <p> The following logic is used. If the given date is greater than the next working date of the Bank, thats aValidation error. If the cutoff has not completed for previous working date for the branch, thats a validation error. @param batchOperation the <tt>BatchOperation</tt> object.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchOpsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchOpsAdapter	getEodBodCutOffStatus	<p>&lt;p&gt;This method returns the status for EOD, BOD and Cut-Off indicating whether each one completed successfully or not. @param sessionContext the session context object. @param date the date for which the status is required. @return the &lt;tt&gt;EodBodCutOffStatusResponse&lt;/tt&gt; object. @throws FatalException</p>	<p>&lt;p&gt;This method returns the status for EOD, BOD and Cut-Off indicating whether each one completed successfully or not. @param sessionContext the session context object. @param date the date for which the status is required. @return the &lt;tt&gt;EodBodCutOffStatusResponse&lt;/tt&gt; object. @throws FatalException</p>
ILoanStatementPrintAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanStatementPrintAdapter	printLoanStatement	<p>&lt;p&gt;This method has been defined to print the statement being generated. &lt;/p&gt; @param sessionContext @param statementPrintingDTO @return</p>	<p>&lt;p&gt;This method is responsible for printing Loan Statements.&lt;br&gt; On Statement Generation Screen, when a request for Adhoc Statement Generation / Re-Ordered Statement Generation is raised, &lt;br&gt; the Statement Details are generated and stored.&lt;br&gt; Once the Statement details are generated, the PRINT option becomes available for user to print the generated items. &lt;/p&gt; @param sessionContext @param loanStatementRequestDTO @return @throws FatalException</p>
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEvent	<p>The method is used to reverse the accounting event raised so far for with the given original transaction reference number. See {@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param transactionReferenceNumber @param originalTransactionReferenceNumber</p>	<p>This method is to reversed the already raised accounting entries for a transaction. All the accounting entries raised against that transaction will be reversed back. @param transactionReferenceNumber @param originalTransactionReferenceNumber</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEvent	The method is used to reverse the accounting event raised so far for with the given original transaction reference number. See {@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param transactionReferenceNumber @param originalTransactionReferenceNumber	This method is to reversed the already raised accounting entries for a transaction. All the accounting entries raised against that event only reversed. @param transactionReferenceNumber @param originalTransactionReferenceNumber @param eventSubSequenceNumber
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEvent	The method is used to reverse the accounting event raised so far for with the given original transaction reference number. See {@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param transactionReferenceNumber @param originalTransactionReferenceNumber	This method reverse all the transaction done during settlement payout instruction execution. It includes 1) Contribution 2) All Fees 3) Legal LMI Fee 4) Payout The debit instruction reversal flow works as follows. <ol> <li>It will load debit instruction by using instruction id and account id</li> <li>Now by using obtained debit instruction it will determine transaction reference number.</li> <li>From given transaction reference number it will determine list of {@link TransactionAuditItem} to process</li> <li>In account reversal system always perform reversal of debit transaction first so amount is always available for reversal of credit</li> <li>So system reverses the fees charged, it also reverses the LMI and Legal fees charged on the system</li> <li>Then it has to fund the amount which is disbursed from this account.</li> <li>Now whatever accounting is performed for original transaction is reversed by using {@link IDomainAccountingAdapter#raiseReverseAccountingEventForTransaction(String, String, String)}</li> <li>Now it will reverse the contribution done on the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account.The contribution reversal is separate forward transaction so it has to raise the accounting for that.</li> <li>So it calls {@link #processAccounting(DebitInstruction, SourcingResponse)} for processing accounting for that transaction.</li> </ol> @param instructionID @param accountId @param originalTransactionReferenceNumber @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEvent	The method is used to reverse the accounting event raised so far for with the given original transaction reference number. See {@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param transactionReferenceNumber @param originalTransactionReferenceNumber	The method is used to raise reverse financial event.{@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param sessionContext
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEvent	The method is used to reverse the accounting event raised so far for with the given original transaction reference number. See {@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param transactionReferenceNumber @param	The method is used to raise reverse financial event.{@link DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)} @param sessionContext

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			originalTransactionReferenceNumber	
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEventForTransaction	The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus</a> . <ol style="list-style-type: none"> <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </ol> @ <a href="#">param transactionReferenceNumber</a> @ <a href="#">param originalTransactionReferenceNumber</a>	This method is to reversed the already raised accounting entries for a transaction. All the accounting entries raised against that transaction will be reversed back. @ <a href="#">param transactionReferenceNumber</a> @ <a href="#">param originalTransactionReferenceNumber</a>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEventForTransaction	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">{@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus}</a>.  <ol style="list-style-type: none"> <li>AccountingBehaviourDTO has two attributes <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </li></ol> </p> <p><a href="#">@param transactionReferenceNumber</a>  <a href="#">@param originalTransactionReferenceNumber</a></p>	<p>This method reverse all the transaction done during settlement payout instruction execution. It includes 1) Contribution 2) All Fees 3) Legal LMI Fee 4) Payout The debit instruction reversal flow works as follows.</p> <ol style="list-style-type: none"> <li>It will load debit instruction by using instruction id and account id</li> <li>Now by using obtained debit instruction it will determine transaction reference number.</li> <li>From given transaction reference number it will determine list of <a href="#">{@link TransactionAuditItem}</a> to process</li> <li>In account reversal system always perform reversal of debit transaction first so amount is always available for reversal of credit</li> <li>So system reverses the fees charged, it also reverses the LMI and Legal fees charged on the system</li> <li>Then it has to fund the amount which is disbursed from this account.</li> <li>Now whatever accounting is performed for original transaction is reversed by using <a href="#">{@link IDomainAccountingAdapter#raiseReverseAccountingEventForTransaction(String, String, String)}</a></li> <li>Now it will reverse the contribution done on the account. The contribution reversal is separate forward transaction so it has to raise the accounting for that.</li> <li>So it calls <a href="#">{@link #processAccounting(DebitInstruction, SourcingResponse)}</a> for processing accounting for that transaction.</li> </ol> <p><a href="#">@param instructionID</a>  <a href="#">@param accountId</a>  <a href="#">@param originalTransactionReferenceNumber</a> <a href="#">@return</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseReverseAccountingEventForTransaction	The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber:String, inputDTO:AccountingBehaviourDTO)::TransactionStatus</a> . <ol style="list-style-type: none"> <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </ol> @params transactionReferenceNumber originalTransactionReferenceNumber	The method is used to raise reverse financial event. <a href="#">DomainAccountingAdapter#raiseReverseAccountingEvent(transactionReferenceNumber: String, originalTransactionReferenceNumber: String)</a> @params sessionContext
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">DomainAccountingAdapter#raiseAccountingEvent</a> @params accountingEventDTO	This method raise the accounting entries. This method can be called as many times as the caller want. If the caller calls with the same eventId, the sequence number for the same event is incremented to facilitate the accounting framework requirement. If this method is called for the first time for the eventId, then this method get the first sequence number for the event with the existing transaction reference number. @params accountingEventAmountTagDetailsDTOList transactionReferenceNumber transactionDescription primaryAccountId @params eventId @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">{@link DomainAccountingAdapter#raiseAccountingEvent}</a> @param accountingEventDTO	This method is used to withdraw cash from a given demand deposit account identified by cashTransaction.accountTransaction.accountId. The withdrawal count is increased by one each time there is call to this transaction. It also gets decreased by one if the transaction is reversed. The cash withdrawal method works as follows :  <ol> <li>Convert the CashTransactionDTO to CashTransaction using <a href="#">{@link CashTransactionAssembler#toInformationObject}</a> (cashTransactionDTO: CashTransactionDTO): : CashTransaction}  <li>Check business policies <ol> The business policy for the cash deposit needs to be qualified. Please refer to each method for explanation of the validation. <li> <a href="#">{@link FinancialTransactionBusinessPolicy#validateAccountNo}</a> </li> <li> <a href="#">{@link FinancialTransactionBusinessPolicy#validateTransactionAmount}</a> </li> <li> <a href="#">{@link FinancialTransactionBusinessPolicy#validateInstrument}</a> </li> <li> <a href="#">{@link FinancialTransactionBusinessPolicy#validateInstrumentDate}</a> </li> <li> <a href="#">{@link CashWithdrawalBusinessPolicy#validateTransactionRepresentative}</a> </li> <li> <a href="#">{@link CashWithdrawalBusinessPolicy#validateDenomination}</a> </li> </ol> </li> <li>call super.manageTransactionDates(transactionDates: TransactionDates) for postingDate validation.</li> <li>Get transactionReferenceNumber: Set the value of CashTransaction.AccountTransaction.transactionReferenceNumber from already generated external reference number in SessionContext().</li> <li>Update account balance: Call <a href="#">{@link DemandDepositAccountService#debitFunds}</a> (accountTransaction: AccountTransaction): DebitResponse} for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>debiting the account with the tranaction amuont.&lt;/li&gt;  &lt;li&gt;The FeeManagementService will be called for calculating and charging the fees i.e.{@link super.manageFees(accountTransaction: AccountTransaction, feeDetails: FeeDetails, instrumentId: String)}&lt;/li&gt; &lt;li&gt;Log financial accounting event using {@link super#raiseFinancialEvents(accountTransaction: AccountTransaction, feeDetails: FeeDetails, instrumentId: String)}&lt;/li&gt; &lt;/ol&gt; &lt;li&gt;Check override conditions &lt;ol&gt; There is another business policy to check if the transaction has utilized the additional amounts which also known as override conditions. {@link handleDebitOverrides(debitResponses: DebitResponse[])} </p>
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See {@link DomainAccountingAdapter#raiseAccountingEvent} @param accountingEventDTO	@param transaction @param txnAmountInAcy @param suspenseAmountInAcy @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	@param transaction @param txnAmountInAcy @param suspenseAmountInAcy @throws FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	This method is used to deposit cash from a given demand deposit account identified by cashTransaction.accountTransaction.accountId. The financial method works as follows: <ol style="list-style-type: none"> <li>Call <a href="#">{@link getInternalTransactionReferenceNumber}</a>: internalTransactionReferenceNumber</li> <li>validate existence of the transaction by super.validateDuplicateTransaction(internalTransactionReferenceNumber: String, SessionContext.externalReferenceNumber: String)</li> <li>Convert the CashTransactionDTO to CashTransaction using <a href="#">{@link CashTransactionAssembler#toInformationObject}</a> (cashTransactionDTO: CashTransactionDTO): : CashTransaction</li> <li>Create FeeDetails object and set FeeDetails.FeeAmount=CashTransaction.FeeAmount and FeeDetails.PriceTicketNumber = CashTransaction.PriceTicketNumber</li> <li>Set FinancialEvent as: accountTransaction.FinancialEventType=FinancialEventType.Regular</li> <li>Set eventId as: accountTransaction.TransactionAttributes.EventId="CASA001"</li> <li>Set internalTransactionReferenceNumber as:</li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>accountTransaction.TransactionReferenceNumber=internalTransactionReferenceNumber&lt;/li&gt; &lt;li&gt;Check business policies by calling {@link checkCashDepositPolicies(cashTransaction: CashTransaction)} &lt;ul&gt; The business policy for the cash deposit needs to be qualified. Please refer to each method for explanation of the validation. &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateAccountNo()}&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateSourceCurrency()}&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateSourceCurrencyAmount()}&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateTransactionBranch()}&lt;/li&gt; &lt;li&gt;{@link CashDepositBusinessPolicy#validateDenomination()}&lt;/li&gt; &lt;li&gt;{@link CashDepositBusinessPolicy#validateTransactionRepresentative()}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;convert the Transaction.AccountTransaction.TransactionAmounts.accountAmount into transactionAmount by calling super.convertToCurrencyAmount(transactionAmounts: TransactionAmounts,sourceCurrency:SourceCurrency)&lt;/li&gt; &lt;li&gt;call super.manageTransactionDates(transactionDates: TransactionDates, transactionBranch: TransactionBranch) for posting date validation.&lt;/li&gt; &lt;li&gt;call {@link DemandDepositAccountService#creditFunds(accountTransaction: AccountTransaction): CreditResponse} for crediting the account with the transaction amount.&lt;/li&gt; &lt;li&gt;The following domain service will be called for calculating and charging the fees.&lt;br&gt;{@link super.manageFees(accountTransaction:</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>AccountTransaction, feeDetails: FeeDetails, ModuleType.CASA)}&lt;/li&gt; &lt;li&gt;Log financial accounting event using {@link super#raiseFinancialEvents(accountTransaction: AccountTransaction,demandDepositAccount: DemandDepositAccount,debitResponce: DebitResponce, feeDetails: FeeDetails, instrumentAccountId: String, ledgerId: String)}&lt;/li&gt;</p> <ul style="list-style-type: none"> <li>Override condtions &lt;ul&gt; There is another business policy to check if the account or the account holder falls in the override conditions. &lt;li&gt;if customer is blacklisted then it is also comes under the override condition&lt;/li&gt; &lt;li&gt;if there is high/medium severity memo then this is also a override condition&lt;/li&gt; &lt;li&gt;Sms module will have configuration if to ignore/authorize/reject the transaction.&lt;/li&gt; &lt;/ul&gt;</li> </ul> <p>&lt;/li&gt; &lt;/ol&gt; @param cashTransactionDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	This method is wrapper service for DashBoard to deposit cash,this method sets the response with hold amounts,fees amounts set on the account.This method is used to deposit cash from a given demand deposit account identified by cashTransaction.accountTransaction.accountId. The financial method works as follows: <ol style="list-style-type: none"> <li>Call <a href="#">getInternalTransactionReferenceNumber</a> to get the internalTransactionReferenceNumber</li> <li>validate existence of the transaction by super.validateDuplicateTransaction(internalTransactionReferenceNumber: String, SessionContext.externalReferenceNumber: String)</li> <li>Convert the CashTransactionDTO to CashTransaction using <a href="#">CashTransactionAssembler#toInformationObject</a> (cashTransactionDTO: CashTransactionDTO): : CashTransaction} <li>Create FeeDetails object and set FeeDetails.FeeAmount=CashTransaction.FeeAmount and FeeDetails.PriceTicketNumber = CashTransaction.PriceTicketNumber</li> <li>Set FinancialEvent as: accountTransaction.FinancialEventType=FinancialEventType.Regular</li> <li>Set EventId as: accountTransaction.TransactionAttributes.EventId="CASA001"</li> <li>Set internalTransactionReferenceNumber as: accountTransaction.TransactionReferenceNumber=internalTransactionReferenceNumber</li> <li>Check business policies by calling <a href="#">checkCashDepositPolicies</a>(cashTransaction: CashTransaction)} <ul style="list-style-type: none"> <li>The business policy for the cash deposit needs to be qualified. Please refer to each method for explanation of the validation.</li> </ul> </li> </li></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FinancialTransactionBusinessPolicy#validateAccountNo()&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateSourceCurrency()}&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateSourceCurrencyAmount()}&lt;/li&gt; &lt;li&gt;{@link FinancialTransactionBusinessPolicy#validateTransactionBranch()}&lt;/li&gt; &lt;li&gt;{@link CashDepositBusinessPolicy#validateDenomination()}&lt;/li&gt; &lt;li&gt;{@link CashDepositBusinessPolicy#validateTransactionRepresentative()}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;convert the Transaction.AccountTransaction.TransactionAmounts.accountAmount into transactionAmount by calling super.convertToCurrencyAmount(transactionAmounts: TransactionAmounts,sourceCurrency:SourceCurrency)&lt;/li&gt; &lt;li&gt;call super.manageTransactionDates(transactionDates: TransactionDates, transactionBranch: TransactionBranch) for posting date validation.&lt;/li&gt; &lt;li&gt;call {@link DemandDepositAccountService#creditFunds(accountTransaction: AccountTransaction): CreditResponse} for crediting the account with the tranaction amuont.&lt;/li&gt; &lt;li&gt;The following domain service will be called for calculating and charging the fees.&lt;br&gt;{@link super.manageFees(accountTransaction: AccountTransaction, feeDetails: FeeDetails, ModuleType.CASA)}&lt;/li&gt; &lt;li&gt;Log financial accounting event using {@link super#raiseFinancialEvents(accountTransaction: AccountTransaction,demandDepositAccount: DemandDepositAccount,debitResponce: DebitResponce, feeDetails: FeeDetails,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> instrumentAccountId: String, ledgerId: String))&lt;/li&gt; &lt;li&gt;Override condtions &lt;ul&gt; There is another business policy to check if the account or the account holder falls in the override conditions. &lt;li&gt;if customer is blacklisted then it is also comes under the override condition&lt;/li&gt; &lt;li&gt;if there is high/medium severity memo then this is also a override condition&lt;/li&gt; &lt;li&gt;Sms module will have configuration if to ignore/authorize/reject the transaction.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ol&gt; @param cashTransactionDTO @param sessionContext @param cashTransactionDTO @return @throws FatalException </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	<p>The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO</p>	<p>This method is wrapper service for DashBoard to withdraw cash,this method sets the response with sweep in amount,fees amount set on the account.This method is used to withdraw cash from a given demand deposit account identified by cashTransaction.accountTransaction.accountId. The withdrawal count is increased by one each time there is call to this transaction. It also gets decreased by one if the transaction is reversed. The cash withdrawal method works as follows :&lt;br&gt; &lt;ol&gt; &lt;li&gt;Convert the CashTransactionDTO to CashTransaction using <a href="#">{@link CashTransactionAssembler#toInformationObject (cashTransactionDTO: CashTransactionDTO): : CashTransaction}</a>&lt;br&gt; &lt;li&gt;Check business policies&lt;br&gt; &lt;ol&gt; The business policy for the cash deposit needs to be qualified. Please refer to each method for explanation of the validation. &lt;li&gt;<a href="#">{@link FinancialTransactionBusinessPolicy#validateAccountNo}</a>&lt;/li&gt; &lt;li&gt;<a href="#">{@link FinancialTransactionBusinessPolicy#validateTransactionAmount}</a>&lt;/li&gt; &lt;li&gt;<a href="#">{@link FinancialTransactionBusinessPolicy#validateInstrument}</a>&lt;/li&gt; &lt;li&gt;<a href="#">{@link FinancialTransactionBusinessPolicy#validateInstrumentDate}</a>&lt;/li&gt; &lt;li&gt;<a href="#">{@link CashWithdrawalBusinessPolicy#validateTransactionRepresentative}</a>&lt;/li&gt; &lt;li&gt;<a href="#">{@link CashWithdrawalBusinessPolicy#validateDenomination}</a>&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;call super.manageTransactionDates(transactionDates: TransactionDates) for postingDate validation.&lt;/li&gt; &lt;li&gt;Get transactionReferenceNumber: Set the value of CashTransaction.AccountTransaction.transactionReferenceNumber from already generated external reference number in SessionContext().&lt;/li&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;Update account balance: Call <a href="#">{@link DemandDepositAccountService#debitFunds(accountTransaction: AccountTransaction): DebitResponse}</a> for debiting the account with the transaction amount.&lt;/li&gt;</p> <p>&lt;li&gt;The FeeManagementService will be called for calculating and charging the fees i.e. <a href="#">{@link super.manageFees(accountTransaction: AccountTransaction, feeDetails: FeeDetails, instrumentId: String)}</a>&lt;/li&gt;</p> <p>&lt;li&gt;Log financial accounting event using <a href="#">{@link super#raiseFinancialEvents(accountTransaction: AccountTransaction, feeDetails: FeeDetails, instrumentId: String)}</a>&lt;/li&gt;</p> <p>&lt;/ol&gt;</p> <p>&lt;li&gt;Check override conditions</p> <p>&lt;ol&gt;</p> <ul style="list-style-type: none"> <li>There is another business policy to check if the transaction has utilized the additional amounts which also known as override conditions. <a href="#">{@link handleDebitOverrides(debitResponses: DebitResponse[])}</a></li> <li>if the amounts like OD, OV and TOD is greater than zero that means the limit is utilized so becomes candidate for override condition&lt;/li&gt;</li> <li>if customer is blacklisted then it is also comes under the override condition&lt;/li&gt;</li> <li>if there is high/medium severity memo then this is also a override condition&lt;/li&gt;</li> </ul> <p>&lt;li&gt;Sms module will have configuration if to ignore/authorize/reject the transaction.&lt;/li&gt;</p> <p>&lt;/ol&gt;</p> <p>&lt;/li&gt;</p> <p>@param sessionContext @param cashTransactionDTO</p>
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">{@link DomainAccountingAdapter#raiseAcc</a>	@param accountTransaction @param debitResponse @param demandDepositAccount @param instrumentId @param amountTagId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			ountingEvent} @param accountingEventDTO	
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See {@link DomainAccountingAdapter#raiseAccountingEvent} @param accountingEventDTO	@param accountTransaction @param debitResponse @param demandDepositAccount @param instrumentId @param amountTagId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See {@link DomainAccountingAdapter#raiseAccountingEvent} @param accountingEventDTO	@param accountingEventDetailDTO @param feeDetails @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See {@link DomainAccountingAdapter#raiseAccountingEvent} @param accountingEventDTO	The method will credit a CASA account provided, a the specific ledger as requested. It is meant mainly for miscellaneous credits to a CASA account against a GL. It will also be called from file upload transactions like salary credit where normally the salary credited will be from some GL. The method accepts the CASA account, the credit amount, the GL and the batchCode for the transaction. BatchCode is an optional input. IF sent as null, a new JournalBatchEntry will be created and used. It calls {@link JournalEntryBatchDomService#createJournalBatch} over an adapter to perform the credit. @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI, String, accountId, 1-40, accountId of the account which is to be credited. @fcb.param MI, MoneyDTO, amount, , the amount to be credited. @fcb.param MI, String, ledgerCode, 1-20, ledgerCode of the ledger to be hit @fcb.param I, String, batchCode, , The batchCode. It is non-mandatory. If sent as null, a new JournalBatchEntry will be created and then used. @return @throws com.ofss.fc.infra.exception.FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See {@link DomainAccountingAdapter#raiseAccountingEvent} @param accountingEventDTO	@param transaction @param txnAmountInAcy @param suspenseAmountInAcy @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">{@link DomainAccountingAdapter#raiseAccountingEvent}</a> @param accountingEventDTO	The method will debit a CASA account provided, hitting a specific ledger as requested. It is meant mainly for miscellaneous debits to a CASA account against a GL. It will also be called from file upload transactions like salary credit where normally the employer CASA account will be debited and the amount put in some temp GL. The method accepts the CASA account, the debit amount, the GL and the batchCode for the transaction. BatchCode is an optional input. IF sent as null, a new JournalBatchEntry will be created and used. It calls <a href="#">{@link JournalEntryBatchDomService#createJournalBatch}</a> over an adapter to perform the credit. @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI, String, accountId, 1..40, accountId of the account which is to be debited. @fcb.param MI, MoneyDTO, amount, , the amount to be debited. @fcb.param MI, String, ledgerCode, 1..20, ledgerCode of the ledger to be hit @fcb.param I, String, batchCode, , The batchCode. It is non-mandatory. If sent as null, a new JournalBatchEntry will be created and then used. @return @throws com.ofss.fc.infra.exception.FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">{@link DomainAccountingAdapter#raiseAccountingEvent}</a> @param	@param transaction @param txnAmountInAcy @param suspenseAmountInAcy @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			accountingEventDTO	
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	The method is used to raise financial event. <a href="#">{@link AccountingEventApplicationService#raiseNewAccountingEvent(SessionContext, AccountingEventDTO)}</a> @param sessionContext @throws FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	This method used to raise accounting event for credit or debit interest settlement. This method performs following activities. <ol style="list-style-type: none"> <li>Prepare AccountingTransactionEvent with following details <ol style="list-style-type: none"> <li>accountTransaction : <code>input.accountTransaction</code></li> <li>relatedDemandDepositAccount : <code>input.account.getAccountId</code></li> <li>partyId: <code>input.account.getPartyId</code></li> </ol> </li> <li>If the alternate account was not present during the backdated value date, the accounting would be raised with parent account.</li> <li>Call <a href="#">@link IDomainAccountingAdapter#raiseAccountingEvent</a>.</li> <li>TODO: Accounting adapter is in progress,Add attributes as per requirement. </li> </ol> @param iDemandDepositAccount @param accountTransaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @param accountingEventDTO	This method is used to raise the accounting event for the transfer of refund of fees @param accountId : Indicates the parent in case the refund is done on alternate account, otherwise it is the self. @param accountingEventAmountTagDetailsDTOs @param reversalDetailDTO @param applicationContext
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	Accounting Module expects Data in the form of <a href="#">{@link AccountingEventAmountTagDetailsDTO}</a> which contains the details of the transaction of a single leg. This method prepares this DTO from the Debit Response. @param accountTransaction @param debitResponse @param demandDepositAccount @param accountAmountTagSeqNoList @param amountTagId @param isCashTransaction @param instrumentId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	Accounting Module expects Data in the form of <a href="#">{@link AccountingEventAmountTagDetailsDTO}</a> which contains the details of the transaction of a single leg. This method prepares this DTO from the Credit Response. @param accountTransaction @param creditResponse @param demandDepositAccount @param accountAmountTagSeqNoList @param amountTagId @param isCashTransaction @param instrumentId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	This method populates the Transaction amounts to the <a href="#">{@link AccountingEventAmountTagDetailsDTO}</a> . @param accountingTransactionAmount @param accountAmountTagSeqNoList @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	@param accountTransaction @param debitResponse @param demandDepositAccount @param instrumentId @param amountTagId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	@param accountTransaction @param debitResponse @param demandDepositAccount @param instrumentId @param amountTagId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	NGPR2ACI-927: Introduced this method to handle back date value date. This method will be used in case of backdated transaction. This method logic is same as <a href="#">{@link #populateAccountingEventAmountDebit(AccountTransaction, DebitResponse, IDemandDepositAccount, Map, String, boolean, String)}</a> . Only difference is while fetching balances casa module requires value to be same or greater than process date. It is achieved in terms of newTransactionDates variable. @param accountTransaction @param debitResponse @param demandDepositAccount @param instrumentId @param amountTagId @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	@param feeAmount @param accountAmountTagSeqNoList @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	@param accountingTransactionAmount @param accountAmountTagSeqNoList @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	@param feeDetails @throws FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	This method used to raise accounting event for credit or debit interest settlement. This method performs following activities. <ol> <li>Prepare AccountingTransactionEvent with following details <ol> <li>accountTransaction : {@code input.accountTransaction}</li><li>relatedDemandDepositAccount : {@code input.account.getAccountId} <li>partyId: {@code input.account.getPartyId} </li></ol><li>If the alternate account was not present during the backdated value date, the accounting would be raised with parent account.</li></ol> <li>Call {@link IDomainAccountingAdapter#raiseAccountingEvent}.</li><li>TODO: Accounting adapter is in progress,Add attributes as per requirement. </ol> @param iDemandDepositAccount @param accountTransaction
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForTransactionAmount	@param accountingEventDTO @param accountAmountTagSeqNoList	This method generates the AccountingEventAmountTagDetailsDTO[] for account and GL @param accountingEventAmountTagDetailsDTO @param reversalDetailDTO @return AccountingEventAmountTagDetailsDTO[]
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	fetchCASAAmountTagDetailsDTO		@param accountId @param transactionDates @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	fetchCASA AmountTag DetailsDTO		@param demandDepositAccount @param transactionDates @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	fetchCASA AmountTag DetailsDTO		@param accountId @param transactionDates @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForFeeAmount	@param accountingRequestDTO @param debitResponse @param accountAmountTagSeqNoList @return	
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForFeeAmount	@param accountingRequestDTO @param debitResponse @param accountAmountTagSeqNoList @return	This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForFeeAmount	<p>@param accountingRequestDTO            @param debitResponse @param accountAmountTagSeqNoList            @return</p>	<pre>&lt;pre&gt;This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt;Get the alternate account for fees posting for the account on which transaction is being posted. &lt;/li&gt; &lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id. &lt;/li&gt; &lt;li&gt;If alternate account is maintained, it will be returned else original account is returned. The fees will be posted to the account returned. &lt;/li&gt; &lt;li&gt;Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()} &lt;/li&gt; &lt;li&gt;The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them. &lt;/li&gt; &lt;li&gt;Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off. &lt;/li&gt; &lt;li&gt;The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object &lt;/li&gt; &lt;li&gt;Instantiate PricingAdapterFactory. &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter} &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				deducted.</li> </li>Return boolean value indicating success or failure to the caller module.</li> </ol> The following are the parameters taken by the method along with their significance </li> </li>@param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.</li> </li>@return boolean: </li> </li>true: If the fee is deducted and intimated successfully at the pricing end.</li> </li>false: If there is any failure in method execution.</li> </ul> </li> </ul> </li> </li>Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumber} // As there might be more than one legs to financial transaction.</li> </pre> @param feeDetails
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	populateAccountingForFeeAmount	@param accountingRequestDTO @param debitResponse @param accountAmountTagSeqNoList @return	@param feeAmount @param accountAmountTagSeqNoList @return
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	getAccountingBehaviourDTO		@param accountingEventDetailDTO @param feeDetails @throws FatalException
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	getAccountingBehaviourDTO		The method is used to raise financial event. {@link AccountingEventApplicationService#raiseNewAccountingEvent(SessionContext, AccountingEventDTO)}} @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDomainAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDomainAccountingAdapter	getAccountingBehaviourDTO		This method is used to raise the accounting event for the transfer of refund of fees @param accountId : Indicates the parent in case the refund is done on alternate account, otherwise it is the self. @param accountingEventAmountTagDetailsDTOs @param reversalDetailDTO @param applicationContext
ICardAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardAccountAdapter	updateBundleSubscription	This method is used to update bundle subscription @param sessionContext @param accountId @param creditCardAccountInputDTO @throws FatalException	This represents the interface containing the interaction services to be used from accounts module by cards module. @author SoundararajanM @version 1.0
ICardAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardAccountAdapter	updateBundleSubscription	This method is used to update bundle subscription @param sessionContext @param accountId @param creditCardAccountInputDTO @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference. After that it will save the current state of simulation. Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardAccountAdapter	updateBundleSubscription	This method is used to update bundle subscription @param sessionContext @param accountId @param creditCardAccountInputDTO @throws FatalException	The method amendCreditCard is responsible for amend credit card, invoked from origination flow   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also interact with the external card system and amend the card limit.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICardAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardAccountAdapter	updateBundleSubscription	This method is used to update bundle subscription @param sessionContext @param accountId @param creditCardAccountInputDTO @throws FatalException	The method createOrUpdateCreditCardAccountDetails is exposed for external system to create or update credit card account in OBP in async flow  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardAccountAdapter	updateBundleSubscription	This method is used to update bundle subscription @param sessionContext @param accountId @param creditCardAccountInputDTO @throws FatalException	
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method performs the operation of deduplication while onboarding new party. It verifies whether any existing party in the system have a close match with the new party details. If any matches found it will populate those information and make it available to the user. @param sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. The information is used by the system for authentication purposes. @param partyOnboardingdto, , Contains the party information captured for the new party. Based on the details provided the de-dupe check will be performed. Details such as name, phone number, email, etc. will be used for dedupe.
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method publishes the new party information to OCH through integration adapters. @param sessionContext @param dto @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method is used to publish party to party relationships to OCH @param sessionContext @param dto @return TransactionStatus
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method is used to publish party to account relationships to OCH @param sessionContext @param dto @return TransactionStatus
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	There are instances where for a single party multiple party IDs gets created, due to multiple origination system in banks. Such parties are merged after deduplication check. After merging of IDs, one is retained and one is removed from the system. Retained party is survivor and the one which gets removed is victim.  This method publishes the final modified details of survivor to OCH after the merge of partyIds. @param sessionContext @param MDMPartyMergeDTO @return transactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	There are instances when two parties undergo several validation checks to see if due to multiple origination systems, the same party has several accounts. When the validations for merge fail and its concluded that the parties under consideration are different, This method is called to publish the two parties as different entities to MDM. @param sessionContext @param mdmPartyMergeDTO @return @throws FatalException
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method populate party to account relationship data and invokes  ChangeAccountTitleAdapter to publish details to OCH. @throws Throwable

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method populate KYC details and invokes UpdatePartyKYCDetailsAdapter  to publish kyc details to MDM. @throws Throwable
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	<p>&lt;p&gt;This service searches for a party given a set of search criteria.&lt;br/&gt;This service will search for parties from various sources depending on the adapter that is implemented for &lt;b&gt; {@link com.ofss.fc.common.integration.config.IntegrationAdapters#party_service_core_PartySearchApplicationService_searchParty} &lt;/b&gt;. &lt;br/&gt; This adapter dictates whether internal or external source is responsible for getting details for the search results. NullAdapter is implementation of this interface where integration end points are absent. The input search criteria {@link com.ofss.fc.app.search.dto.SearchCondition#searchType} defines the type of party search intended. Any one of the following values can be supplied in this variable: &lt;dl&gt; &lt;dt&gt;PrimarySearchDetails&lt;/dt&gt; &lt;dd&gt;For basic search conditions like partyId, partyType, partyClass (Refer {@link com.ofss.fc.domain.party.service.dto.search.PrimarySearchDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;IdentificationDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of identification details like identificationType, identificationValue, issueDate, expiryDate (Refer {@link com.ofss.fc.domain.party.service.dto.search.Identificat</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ionDetails} for complete list of input criteria)&lt;/dd&gt;  &lt;dt&gt;NameDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of secondary name details like firstName, middleName, shortName, lastName (Refer {@link com.ofss.fc.domain.party.service.dto.search.NameDetails} for complete list of input criteria)&lt;/dd&gt;  &lt;dt&gt;AddressDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of address details like city, state, country (Refer {@link com.ofss.fc.domain.party.service.dto.search.AddressDetails} for complete list of input criteria)&lt;/dd&gt;  &lt;dt&gt;BrokerDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of broker details like spid, serviceProviderStatusType, commissionRecipientId, commissionRecipientName (Refer {@link com.ofss.fc.domain.party.service.dto.search.BrokerDetails} for complete list of input criteria)&lt;/dd&gt;  &lt;dt&gt;RoleDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of role details like roleId (Refer {@link com.ofss.fc.domain.party.service.dto.search.RoleDetails} for complete list of input criteria)&lt;/dd&gt; &lt;/dl&gt; For each of the above search conditions, the user can choose whether the search is to use exact match or not by specifying {@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.isExactSearch}. &lt;i&gt;Default value is true&lt;/i&gt;&lt;br/&gt; For example if the user intends to search on first name, and specifies input as isExactSearch=true and firstName='A', the service will fetch records where the firstName='A' but if isExactSearch='false', the service will fetch records where the firstName like '%a%'&lt;br/&gt; &lt;br/&gt; This transaction supports the Pagination feature. The user can control the number of records to be fetched by specifying {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.app.party.dto.search.input.PartySearchRequest.totalRecordsFetched} <i>Default value is 10</i></p> <p>The user can also specify the first row from which data is sought by setting <a href="#">PartySearchRequest.firstResult</a>. <i>Default value is 0</i></p> <p>The party search results returned in a search contain the basic details of each party found <a href="#">PartySearchResponseDTO</a>. There may be times when additional details are required. To facilitate this, the search transaction supports an inquiry level which the user can specify through <a href="#">PartySearchRequest.isBasicSearch</a> <i>Default value is true</i></p> <p>The response fetches the following details for parties which match the search criteria:</p> <ul style="list-style-type: none"> <li>Party id - unique identifier for the party</li> <li>Party name - full name of the party</li> <li>Party class</li> <li>Party type - indicator of whether the party is Individual, Bank, Trust etc.</li> <li>Date of birth - birth date of the party</li> <li>Email id - email of the party</li> <li>Postal code - address postal code of the party</li> <li>Role id - customer or banker role</li> </ul> <p>@param sessionContext @param partySearchrequest @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IIntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method will fetch party account information from OCH for advances search flow
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	<p>This method fetches party details, party contact points and balance summary of all accounts held by the input party.   <i>OBP Party id is a mandatory input to this service.</i>    This service returns party details including partyId, full name, party type and date of birth, address (communication by default and if not found, home address), party contact points and balance details of the accounts held by the party. The response includes basic details of all the accounts. Basic details include account number, nick name, currency and status. The balance details for the accounts are returned in an array of {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.app.party.dto.relation.account.balances.AccountBalanceDTO} which stores information like BalanceType and the corresponding amount.&lt;br/&gt;The BalanceType returned can be one or more of the following depending on the nature of the account:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Net&lt;/li&gt; &lt;li&gt;Unclear&lt;/li&gt; &lt;li&gt;RPA&lt;/li&gt; &lt;li&gt;Outstanding&lt;/li&gt; &lt;li&gt;Regular&lt;/li&gt; &lt;li&gt;Uncollected&lt;/li&gt; &lt;li&gt;CurrNet&lt;/li&gt; &lt;li&gt;CurrAvailable&lt;/li&gt; &lt;li&gt;CurrOutstanding&lt;/li&gt; &lt;li&gt;AvgAvailLastYear&lt;/li&gt; &lt;li&gt;AvgNetLastYear&lt;/li&gt; &lt;li&gt;WithdrawableBalance&lt;/li&gt; &lt;li&gt;DisbursedAmount&lt;/li&gt; &lt;li&gt;Principal&lt;/li&gt; &lt;li&gt;RunningBalance&lt;/li&gt; &lt;li&gt;SanctionedODAmount&lt;/li&gt; &lt;/ul&gt; &lt;/p&gt; <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication. @fcb.param MI,String, partyId is the party or customer id @return {@link com.ofss.fc.app.party.dto.account.PartyAccountBalancesResponse} @throws FatalException</p> </li></ul>
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	<p>This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto</p>	<p>&lt;p&gt;This method fetches party details, party contact points and balance summary of all accounts held by the input party. &lt;br/&gt; &lt;i&gt;The search can be based on OBP Party id or MDM External party id which uniquely identifies the customer . This is a mandatory input to this service.&lt;/i&gt; &lt;br/&gt; &lt;br/&gt;This service returns party details including partyId, full name, party type and date of birth, address (communication by default and if not found, home address), party contact points and balance details of the accounts held by the party. The response includes basic details of all</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the accounts. Basic details include account number, nick name, currency and status. The balance details for the accounts are returned in an array of {@link com.ofss.fc.app.party.dto.relation.account.balances.AccountBalanceDTO} which stores information like BalanceType and the corresponding amount.&lt;br/&gt;The BalanceType returned can be one or more of the following depending on the nature of the account:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Net&lt;/li&gt;</li> <li>&lt;li&gt;Unclear&lt;/li&gt;</li> <li>&lt;li&gt;RPA&lt;/li&gt;</li> <li>&lt;li&gt;Outstanding&lt;/li&gt;</li> <li>&lt;li&gt;Regular&lt;/li&gt;</li> <li>&lt;li&gt;Uncollected&lt;/li&gt;</li> <li>&lt;li&gt;CurrNet&lt;/li&gt;</li> <li>&lt;li&gt;CurrAvailable&lt;/li&gt;</li> <li>&lt;li&gt;CurrOutstanding&lt;/li&gt;</li> <li>&lt;li&gt;AvgAvailLastYear&lt;/li&gt;</li> <li>&lt;li&gt;AvgNetLastYear&lt;/li&gt;</li> <li>&lt;li&gt;WithdrawableBalance&lt;/li&gt;</li> <li>&lt;li&gt;DisbursedAmount&lt;/li&gt;</li> <li>&lt;li&gt;Principal&lt;/li&gt;</li> <li>&lt;li&gt;RunningBalance&lt;/li&gt;</li> <li>&lt;li&gt;SanctionedODAmount&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;/p&gt;</p> <p>@fcb.param MI.com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication. @fcb.param MI,String, partyId, ,OBP Party id or MDM External party id which uniquely identifies the customer @return {@link com.ofss.fc.app.party.dto.account.PartyAccountBalancesResponse} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	update	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Async strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dto	This method will fetch party account information from OCH and return the balance details of the accounts
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	This method performs the operation of deduplication while onboarding new party. It verifies whether any existing party in the system have a close match with the new party details. If any matches found it will populate those information and make it available to the user. @param sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. The information is used by the system for authentication purposes. @param partyOnboardingdto, , Contains the party information captured for the new party. Based on the details provided the de-dupe check will be performed. Details such as name, phone number, email, etc. will be used for dedupe.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	<p>&lt;p&gt;This service searches for a party given a set of search criteria.&lt;br/&gt;This service will search for parties from various sources depending on the adapter that is implemented for &lt;b&gt; {@link com.ofss.fc.common.integration.config.IntegrationAdapters#party_service_core_PartySearchApplicationService_searchParty} &lt;/b&gt;. &lt;br/&gt; This adapter dictates whether internal or external source is responsible for getting details for the search results. NullAdapter is implementation of this interface where integration end points are absent. The input search criteria {@link com.ofss.fc.app.search.dto.SearchCondition#searchType} defines the type of party search intended. Any one of the following values can be supplied in this variable: &lt;dl&gt; &lt;dt&gt;PrimarySearchDetails&lt;/dt&gt; &lt;dd&gt;For basic search conditions like partyId, partyType, partyClass (Refer {@link com.ofss.fc.domain.party.service.dto.search.PrimarySearchDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;IdentificationDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of identification details like identificationType, identificationValue, issueDate, expiryDate (Refer {@link com.ofss.fc.domain.party.service.dto.search.IdentificationDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;NameDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of secondary name details like firstName, middleName, shortName, lastName (Refer {@link com.ofss.fc.domain.party.service.dto.search.NameDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;AddressDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of address details like city, state, country (Refer {@link com.ofss.fc.domain.party.service.dto.search.AddressDetails} for complete list of input criteria)&lt;/dd&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;dt&gt;BrokerDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of broker details like spid, serviceProviderStatusType commissionRecipientId, commissionRecipientName (Refer {@link com.ofss.fc.domain.party.service.dto.search.BrokerDetails} for complete list of input criteria)&lt;/dd&gt;</p> <p>&lt;dt&gt;RoleDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of role details like roleId (Refer {@link com.ofss.fc.domain.party.service.dto.search.RoleDetails} for complete list of input criteria)&lt;/dd&gt; &lt;/dl&gt; For each of the above search conditions, the user can choose whether the search is to use exact match or not by specifying {@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.isExactSearch}. &lt;i&gt;Default value is true&lt;/i&gt;&lt;br/&gt; For example if the user intends to search on first name, and specifies input as isExactSearch=true and firstName='A', the service will fetch records where the firstName='A' but if isExactSearch='false', the service will fetch records where the firstName like '%a%'&lt;br/&gt; &lt;br/&gt; This transaction supports the Pagination feature. The user can control the number of records to be fetched by specifying {@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.totalRecordsFetched} &lt;i&gt;Default value is 10&lt;/i&gt;&lt;br/&gt; &lt;br/&gt; The user can also specify the first row from which data is sought by setting {@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.firstResult}. &lt;i&gt;Default value is 0&lt;/i&gt; &lt;br/&gt; &lt;br/&gt; The party search results returned in a search contain the basic details of each party found {@link com.ofss.fc.app.party.dto.search.output.PartySearchResponseDTO}. There may be times when additional details are required. To facilitate this, the search</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>transaction supports an inquiry level which the user can specify through {@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.isBasicSearch} <i>&lt;i&gt;Default value is true&lt;/i&gt;</i></p> <p>The response fetches the following details for parties which match the search criteria:</p> <ul style="list-style-type: none"> <li>Party id - unique identifier for the party</li> <li>Party name - full name of the party</li> <li>Party class</li> <li>Party type - indicator of whether the party is Individual, Bank, Trust etc.</li> <li>Date of birth - birth date of the party</li> <li>Email id - email of the party</li> <li>Postal code - address postal code of the party</li> <li>Role id - customer or banker role</li> </ul> <p>@param sessionContext @param partySearchrequest @return @throws FatalException</p>
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	<p>This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies</p> <p>@param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well.</p> <p>@param dummy - This is dummy string used to overload the 'update' method</p>	<p>This method will fetch party account information from OCH for advances search flow</p>
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	<p>This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes</p>	<p>This method fetches party details, party contact points and balance summary of all accounts held by the input party.</p> <p><i>&lt;i&gt;OBP Party id is a mandatory input to this service.&lt;/i&gt;</i></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	nAdapter		<p>done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method</p>	<p>&lt;br/&gt;This service returns party details including partyId, full name, party type and date of birth, address (communication by default and if not found, home address), party contact points and balance details of the accounts held by the party. The response includes basic details of all the accounts. Basic details include account number, nick name, currency and status. The balance details for the accounts are returned in an array of {@link com.ofss.fc.app.party.dto.relation.account.balances.AccountBalanceDTO} which stores information like BalanceType and the corresponding amount.&lt;br/&gt;The BalanceType returned can be one or more of the following depending on the nature of the account:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Net&lt;/li&gt;</li> <li>&lt;li&gt;Unclear&lt;/li&gt;</li> <li>&lt;li&gt;RPA&lt;/li&gt;</li> <li>&lt;li&gt;Outstanding&lt;/li&gt;</li> <li>&lt;li&gt;Regular&lt;/li&gt;</li> <li>&lt;li&gt;Uncollected&lt;/li&gt;</li> <li>&lt;li&gt;CurrNet&lt;/li&gt;</li> <li>&lt;li&gt;CurrAvailable&lt;/li&gt;</li> <li>&lt;li&gt;CurrOutstanding&lt;/li&gt;</li> <li>&lt;li&gt;AvgAvailLastYear&lt;/li&gt;</li> <li>&lt;li&gt;AvgNetLastYear&lt;/li&gt;</li> <li>&lt;li&gt;WithdrawableBalance&lt;/li&gt;</li> <li>&lt;li&gt;DisbursedAmount&lt;/li&gt;</li> <li>&lt;li&gt;Principal&lt;/li&gt;</li> <li>&lt;li&gt;RunningBalance&lt;/li&gt;</li> <li>&lt;li&gt;SanctionedODAmount&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;/p&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication. @fcb.param MI,String, partyId is the party or customer id @return {@link com.ofss.fc.app.party.dto.account.PartyAccountBalancesResponse} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	<p>This method fetches party details, party contact points and balance summary of all accounts held by the input party. The search can be based on OBP Party id or MDM External party id which uniquely identifies the customer. This is a mandatory input to this service. This service returns party details including partyId, full name, party type and date of birth, address (communication by default and if not found, home address), party contact points and balance details of the accounts held by the party. The response includes basic details of all the accounts. Basic details include account number, nick name, currency and status. The balance details for the accounts are returned in an array of <code>{@link com.ofss.fc.app.party.dto.relation.account.balances.AccountBalanceDTO}</code> which stores information like BalanceType and the corresponding amount. The BalanceType returned can be one or more of the following depending on the nature of the account:</p> <ul style="list-style-type: none"> <li>Net</li> <li>Unclear</li> <li>RPA</li> <li>Outstanding</li> <li>Regular</li> <li>Uncollected</li> <li>CurrNet</li> <li>CurrAvailable</li> <li>CurrOutstanding</li> <li>AvgAvailLastYear</li> <li>AvgNetLastYear</li> <li>WithdrawableBalance</li> <li>DisbursedAmount</li> <li>Principal</li> <li>RunningBalance</li> <li>SanctionedODAmount</li> </ul> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication. @fcb.param MI,String, partyId, ,OBP Party id or MDM External</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				party id which uniquely identifies the customer @return {@link com.ofss.fc.app.party.dto.account.PartyAccountBalancesResponse} @throws FatalException
IIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\IIntegrationAdapter	updateWithResponse	This method uses appropriate Assembler and IntegrationStrategy to invoke appropriate Adaptee method, to propagate the changes done by the FC business method. This will be implemented as part of Sync strategies @param dto - This is the domain DTO which has been changed by the FC business method. This can be aggregation of DomainDTOs as well. @param dummy - This is dummy string used to overload the 'update' method	This method will fetch party account information from OCH and return the balance details of the accounts
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for CCI Welcome Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for Cover Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for HypothecationLetter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for PersonalGuaranteeLetter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for Mortgage Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchProductGroup	This represents the fetchProductGroup method of the entity. This method fetches an existing product group from the table. @return {@link com.ofss.fc.app.pm.dto.productgroup.ProductGroupInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate SubLine for Main Line of Product Groups. @param context @param facilityId @param loanAccountSublineDTO @param outboundDocumentRequestDTO @return SubLine @throws FatalException
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationPMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAadapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationPMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPMAAdapter	fetchOffer	This represents the fetchOffer method of the entity. @return {@link com.ofss.fc.app.pm.dto.offer.OfferInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTaxationAdapter	inquireTaxDetails		This method will return all the details of a deposit that need to be displayed in the statement inquiry screen. @param sessionContext @param accountId @param depositNumber @param interestClass @param interestDivision @param effectiveDate @throws FatalException
IAccountTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTaxationAdapter	inquireTaxDetails		Method will call the tax inquiry service in { <a href="#">@link GeneralTaxationInquiryApplicationService</a> } and return the response.
IAccountTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTaxationAdapter	fetchTransactionFailureLogs		
IAccountTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTaxationAdapter	fetchTransactionFailureLogs		
IAccountCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountCollectionAdapter	updateHardshipDetails		This method will update collections for all STP and non-STP hardship reliefs requested.
IAccountCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountCollectionAdapter	updateHardshipDetails		This method will interact with collections and update the hardship relief status. @param hardshipReliefRequestDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountCollectionAdapter	fetchHardshipDetailsForAParty		This method will fetch hardship relief application applied by the given party. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param partyId, String @return {@link com.ofss.fc.app.account.dto.hardshiprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountCollectionAdapter	intimateCorrespondancePreferenceUpdate		@param accountAddressLinkage @param accountDDAAdapter @throws FatalException
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	linkAgentDetailsToSubmission		If it is a banker sourced submission, it'll delete all previous agent linkages. In case of an agent sourced submission, it'll link the submission to the agent. @param submissionHeaderDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentDetailsForSubmission		The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology.This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference.After that it will save the current state of simulation.Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders,add-on card holders, account preferences such as balance transfer details,delivery preferences,card membership details and also the preferences for each card holder such as emboss name,card background image @return {@link CreditCardAccountResponse} This class contains account id and account configuration DTO @throws FatalException
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentDetailsForSubmission		This method will delete the agent details linked to the submission



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentDetailsForSubmission		<p>This method is used to open a Lending Account.&lt;br&gt;The following validations are performed:&lt;br&gt; 1. There should not be any warning indicators for the parties applying for the PG&lt;br&gt; 2. The KYC check should be completed for all the parties&lt;br&gt; 3. All the preliminary conditions should have been complied with &lt;br&gt; 4. Productgroup level Origination fees should have been collected &lt;br&gt; 5. Application should be verified successfully.&lt;br&gt; Once all the validations are successful, the facility is confirmed in LCM and the account is created &lt;br&gt; The account id is then updated in Origination&lt;br&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @param MI,FacilityApprovalDataDTO facilityApprovalDataDTO - the facility level approval data @return LendingAccountCreationResponse, this class returns the LendingAccountCreationResponse object, @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This check is part of Pre-create account validation. &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties &lt;br&gt; applying to the product group for which account is being opened. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#CONFIGURATIONS_DO_NOT_EXIST} : FC_OR_AC_0001 &lt;br&gt; This error occurs when account configuration for an offer is not found &lt;br&gt; {@link OriginationErrorConstants#OR_PRE_ACCOUNT_CREATION_VALIDATIONS_FAILED} : FC_OR_AC_0052 &lt;br&gt; This error occurs as a part of pre account creation checks. &lt;br&gt; 1. {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; 2. {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties applying to the product group for which account is being opened. &lt;br&gt; 3. {@link OriginationErrorConstants#PRELIMINARY_CONDITIONS_NON_COMPLIANCE} : FC_OR_OF_0057 &lt;br&gt; Preliminary conditions for the borrowing entity have not been complied .&lt;br&gt; 4. {@link OriginationErrorConstants#OR_UNCOLLECTED_ORIGINATION_FEES} : FC_OR_OF_0051 &lt;br&gt; The product group level pricing fees have not been collected .&lt;br&gt; 5. {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} : FC_OR_OF_0047   Application has not been verified yet     {@link OriginationErrorConstants#OR_LENDING_APPLICATION_INVALID} : FC_OR_ST_0001   This error occurs as a part of state transition checks when the application is not a lending application   
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentDetailsForSubmission		This service will update the HeaderDTO with the agent details if the submission is sourced by an Agent. @param sessionContext @param submissionId @param headerDTO @throws FatalException
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	deleteAgentSubmissionLinkage		If it is a banker sourced submission, it'll delete all previous agent linkages. In case of an agent sourced submission, it'll link the submission to the agent. @param submissionHeaderDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	addAgentArrangementLinkage		<p>The method originateCreditCard is responsible for actually doing credit card request &lt;br&gt; The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference. After that it will save the current state of simulation. Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED. &lt;br&gt;</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountConfigurationDTO, creditCardAccountConfigurationDTO, , This class holds information about the account being setup such as primary card holders, add-on card holders, account preferences such as balance transfer details, delivery preferences, card membership details and also the preferences for each card holder such as emboss name, card background image @return {@link CreditCardAccountResponse} This class contains account id and account configuration DTO @throws FatalException</p>
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	addAgentArrangementLinkage		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	addAgentArrangementLinkage		<p>This method is used to open a Lending Account.&lt;br&gt;The following validations are performed:&lt;br&gt; 1. There should not be any warning indicators for the parties applying for the PG&lt;br&gt; 2. The KYC check should be completed for all the parties&lt;br&gt; 3. All the preliminary conditions should have been complied with &lt;br&gt; 4. Productgroup level Origination fees should have been collected &lt;br&gt; 5. Application should be verified successfully.&lt;br&gt; Once all the validations are successful, the facility is confirmed in LCM and the account is created &lt;br&gt; The account id is then updated in Origination&lt;br&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @param MI,FacilityApprovalDataDTO facilityApprovalDataDTO - the facility level approval data @return LendingAccountCreationResponse, this class returns the LendingAccountCreationResponse object, @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This check is part of Pre-create account validation. &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties &lt;br&gt; applying to the product group for which account is being opened. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#CONFIGURATIONS_DO_NOT_EXIST} : FC_OR_AC_0001 &lt;br&gt; This error occurs when account configuration for an offer is not found &lt;br&gt; {@link OriginationErrorConstants#OR_PRE_ACCOUNT_CREATION_VALIDATIONS_FAILED} : FC_OR_AC_0052 &lt;br&gt; This error occurs as a part of pre account creation checks. &lt;br&gt; 1. {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; 2. {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties applying to the product group for which account is being opened. &lt;br&gt; 3. {@link OriginationErrorConstants#PRELIMINARY_CONDITIONS_NON_COMPLIANCE} : FC_OR_OF_0057 &lt;br&gt; Preliminary conditions for the borrowing entity have not been complied .&lt;br&gt; 4. {@link OriginationErrorConstants#OR_UNCOLLECTED_ORIGINATION_FEES} : FC_OR_OF_0051 &lt;br&gt; The product group level pricing fees have not been collected .&lt;br&gt; 5. {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} : FC_OR_OF_0047   Application has not been verified yet     {@link OriginationErrorConstants#OR_LENDING_APPLICATION_INVALID} : FC_OR_ST_0001   This error occurs as a part of state transition checks when the application is not a lending application   
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchArrangementForAgentArrangementType	This adapter will fetch all the AgebtArrangementLinkages fetched by dsaID for Origination Module Type @param dsaId	
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchArrangementForAgentArrangementType	This adapter will fetch all the AgebtArrangementLinkages fetched by dsaID for Origination Module Type @param dsaId	
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentArrangementLinkagesForMultipleArrangementId	This method will fetch the AgentArrangementLinkages DTO for Arrangement ids given in arrangementIds Response will contain the map of the arrangementId and correspondingAgentArrangementLinkageDTO @param sessionContext @param arrangementIds @return @throws FatalException	This method will update the agent details in header if the submission is sourced by an Agent. @param sessionContext @param applicationSearchResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAgentAdapter	fetchAgentArrangementLinkagesForMultipleArrangementId	This method will fetch the AgentArrangementLinkages DTO for Arrangement ids given in arrangementIds Response will contain the map of the arrangementId and correspondingAgentArrangementLinkageDTO @param sessionContext @param arrangementIds @return @throws FatalException	This method will update the agent details in the {@link BasicApplicationSearchResponse}. @param sessionContext @param response @throws FatalException
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This represents the addHardshipReliefRequest method of the entity. 1)Following steps are carried out in the method: 1)System Constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplcation is called via AccountLoanApdater, LoanHardhsipRelief whose approval type are "Pending" are returned from the called method. 5)If DDAHardsipReliefRequestDetails are present, then DDAHardsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.app.account.dto.hardshiprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the workflow tracker summarized entity. @param sessionContext @param hardshipReliefRequestDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	createWorkflowTrackerSummarizedEntity	This method will create the Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will create the workflow tracker summarized entity. @param sessionContext @param hardshipApplicationId
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	updateAccountVariationWorkflowTrackerSummarizedEntity	This method will update the status of Account Variation Workflow Tracker Summarized Entity. @param workflowTrackerSummarizedEntity DTO	This method will update the process description and the account Id in the workflow tracker summarized entity. @param variationRequestDTO
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	updateAccountHardshipReliefWorkflowTrackerSummarizedEntity	This method will update the status of Account HardshipRelief Workflow Tracker Summarized Entity. @param keyAsString @param accountId @param processDescription	This method will create the workflow tracker summarized entity. @param sessionContext @param hardshipApplicationId
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	deleteWorkflowTrackerSummarizedEntity	This method will delete the Workflow Tracker Summarized Entity. @param keyAsString @param entityName	
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	deleteWorkflowTrackerSummarizedEntity	This method will delete the Workflow Tracker Summarized Entity. @param keyAsString @param entityName	<p>This method will delete the Hardship Relief Request. </p> @param hardshipReliefRequest
IAccountWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountWorkflowAdapter	deleteWorkflowTrackerSummarizedEntity	This method will delete the Workflow Tracker Summarized Entity. @param keyAsString @param entityName	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	createBranchGroupMapping	This method creates and entry in the branch group mapping table - FLX_BATCH_JOB_BRN_GRP_MAPPING.   This is called when a new branch is created or reopened. @param bankCode @param branchCode @param branchGroupCode	This method performs add operation for branch group mapping in the batch table. @param branchParameters : {@link IBranchParameters}
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	isCriticalBatchRunning		This method is used to check if a batch is still in progress and if so, then calendar updation needs to be prohibited. @param MI, bankCode, java.lang.String, This parameter accepts bank code for fetching batch details @param MI, branchCode, java.lang.String, This parameter accepts branch code for fetching batch details @return , isBatchRunning, boolean, returns true if batch is running, else returns false.
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		This service method is used to update the changing parameters of a branch. On click of the update button, all the fields that can change will become editable. On submit, data will be validated and persisted. @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		@param bankCode : String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		This represents the addCldrWeeklyHolidayWork method of the entity. @param MI MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI MI,CldrWeeklyHolidayWorkDTO,cldrWeeklyHolidayWorkDTO, ,cldrWeeklyHolidayWorkDTO TODO: Explain the cldrWeeklyHolidayWorkDTO above @return {@link TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		This represents the updateCldrWeeklyHolidayWork method of the entity. @param MI MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param MI MI,CldrWeeklyHolidayWorkDTO,cldrWeeklyHolidayWorkDTO, ,cldrWeeklyHolidayWorkDTO TODO: Explain the cldrWeeklyHolidayWorkDTO above @return {@link TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		@param bankCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	updateBatchGroupCategoryDates		
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	deleteBranchGroupMapping	The method deletes entry from the branch group mapping table - FLX_BATCH_JOB_BRN_GRP_MAPPING.   This record is deleted when a branch is closed or deleted. @param bankCode @param branchCode @param branchGroupCode	This service method is used to close a particular BranchParameters record. @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceBatchAdapter	deleteBranchGroupMapping	The method deletes entry from the branch group mapping table - FLX_BATCH_JOB_BRN_GRP_MAPPING.   This record is deleted when a branch is closed or deleted. @param bankCode @param branchCode @param branchGroupCode	This method will delete the entries from the branch group mapping in the batch table. @param branchParameters : {@link IBranchParameters}
ILinkedAccountInfoFromCasaAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ILinkedAccountInfoFromCasaAdapter	fetchAccountDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method populates the accounting event dto for raising the accounting event. &lt;br&gt; @fcb.param MI, ApplicationContext, applicationContext &lt;li&gt; {@link com.ofss.fc.framework.context.ApplicationContext}&lt;/li&gt; @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventAmountTagDetailsDTO &lt;li&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}&lt;/li&gt; @fcb.param MI, FundTransferType, fundTransferType &lt;li&gt; {@link com.ofss.fc.enumeration.pc.FundTransferType}&lt;/li&gt; @fcb.param MI, String, eventId @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction &lt;li&gt; {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}&lt;/li&gt; @return {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO} @throws BusinessException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method will raise suspense accounting when reversal transaction to the beneficiary fails.&lt;br&gt; @fcb.param MI, ApplicationContext, applicationContext &lt;li&gt; {@link com.ofss.fc.framework.context.ApplicationContext}&lt;/li&gt; @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction &lt;li&gt; {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}&lt;/li&gt; @fcb.param MI, FundTransferType, fileType &lt;li&gt; {@link com.ofss.fc.enumeration.pc.FundTransferType}&lt;/li&gt; @fcb.param MI, String, ledgerCode, Indicates the intermediary hold ledger for business transfer transaction. @fcb.param MI, boolean, isFirstCall &lt;li&gt;Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.&lt;/li&gt; @throws BusinessException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method will validate the Direct Entry Identification Number for the provider account.</p> <p>&lt;br&gt; @param accountCode @param direction @param inputDirectEntryCode</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>Validating Account number by IMC call.</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. @fcb.param transactionLog @fcb.param accountingEventAmountTagDetailsDTOArray @throws FatalException @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>&lt;p&gt;This method fill update the trace account details in the log.&lt;br/&gt; If the trace account id and module is not supplied, it will be defaulted to the provider account id and module. &lt;br/&gt; If trace account id is supplied but trace account module is null, &lt;br/&gt; &lt;ul&gt;&lt;li&gt;if the trace account id is same as provider account id, the trace account module will be set to the fetched account module.&lt;/li&gt; &lt;li&gt;if the trace account id is not same as provider account id, the account module for the input trace account id will be fetched.&lt;/li&gt; &lt;/ul&gt;</p> <p>&lt;/p&gt; @param inputInternalAccountDetail @param fetchedProviderAccountModule @param internalAccountDetail @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>{@link OutgoingElectronicCollectionService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method will do refusal accounting for dishonoured refused transactions and reverse the accounting entries as it will DEBIT end-point and CREDIT suspense @see com.ofss.fc.app.pc.core.AbstractInwardFundsExchangeService#doReturnAccounting(com.ofss.fc.domain.pc.entity .log.TransactionLog)</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>&lt;p&gt; Validating Account number by IMC call. &lt;/p&gt; @param sessionContext @param accountNo @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<pre>{@link OutgoingClearingService#triggerAccounting(TransactionContext, TransactionLog, AccountingEventAmountTagDetailsDTO[], MessageLogItem)}</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	Sends alert to customer for Incoming collection request on his account. @param transactionLog

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method populates the accounting event dto.</p> <p>@param applicationContext @param eventBillPay @param remitComponentPreferenceItemDTO @param listAccountingEventAmountTagDetailsDTO @param transactionAmount @param transactionDescription @param isNotConsolidated @return @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method populates AccountingEventDTO which is required to raise accounting event. @fcb.param MI, TransactionContext, transactionContext &lt;li&gt;{@link com.ofss.fc.app.context.TransactionContext}&lt;/li&gt; @fcb.param MI, TransactionLogDTO, transactionLog &lt;li&gt;{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}&lt;/li&gt; @fcb.param MI, List, accountingEventAmountTagDetailsDTO @fcb.param MI, String, eventId @fcb.param MI, MessageLogItemDTO, messageLogItem &lt;li&gt;{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}&lt;/li&gt; @return PCAccountingEventDTO &lt;li&gt;{@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}&lt;/li&gt; @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method returns Account BSB value @return String
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This represents the addBillPaymentSchedule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This represents the wrapper service for addBillPaymentSchedule method of the entity to be used by dashboard. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method performs add operation. @param counterparty @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method is used for updating the workitem with the party Id. @param billPayDTO @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method is validating the account number. @param sessionContext @param accountNo
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method validates the account number. @param sessionContext @param accountNo @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This service populate the bank draft basic details.&lt;br&gt;&lt;li&gt;It also derives the transaction branch and the payable at institution for the bank draft.&lt;br&gt; @param applicationContext @param bankDraft @return @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>@param log @param taskModuleType @param internalAccountDetailDTO @param internalAccountDetail @return @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method returns Account BSB value @return String
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	This method will check if for a given product, CFC accounting is required or not. @param transactionLog @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	This method will transfer funds with in own bank accounts @param fundTransferDTO @param transactionLog @param messageLogItem @param transactionContext @param messageIdentification @return @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	This method builds the Account Transaction DTO for the case of Business Transfer. </br> @param transactionLog @param messageLogItem @return @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> @param accountId @param accountModuleType @return @throws FatalException	This method populates the accounting event dto for return transactions. @param transactionContext @param transactionLog @param eventId @param ledgerId @param eventAmountTagDetailsDTOs @param messageLogItem TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method populates the accounting event dto for return transactions. @param transactionContext @param transactionLog @param ledgerId @param eventAmountTagDetailsDTOs @param messageLogItem TODO @return @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	<p>This method will fetch very basic details for the supplied customer account and if details are not found it will throw an error. @param instrumentTransferTransactionLog @param internalAccountDetailDTO @param internalAccountDetail @return @throws FatalException</p>
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchAccountDetail	<p>&lt;p&gt;This method will fetch account details given the account id. If the task module type is supplied, it will call the account module directly, else it will call party with the given account id to first determine the account type and then call the corresponding account module.</p> <p>&lt;/p&gt; @param accountId @param accountModuleType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	This service will call the settlement with the supplied componentPreferenceDTO in the bankDraftDTO. <li>This method will first derive the settlement mode type from the ComponentPreferenceDTO. <li>It will also derive the fee(if applicable)for the liquidation of the bank draft.  <li>If liquidation of bank draft is happening through settlement then it will only derive the settlement mode of liquidation. <li>The retrieving of fee and settlement processing will be skipped if the bank draft is liquidated via settlement. @param transactionContext @param eventId @param isSettlementSrc @param featureDTO @param bankDraftDTO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	The procureFunds service will call the settlement with the supplied componentPreferenceDTO in the bankDraftDTO.  <li>This method will first derive the settlement mode type from the ComponentPreferenceDTO. <li>It will also derive the fee(if applicable)for the issuance of the bank draft.  <li>If issuance of bank draft is happening through settlement then it will only derive the settlement mode of issuance. <li>The retrieving of fee and settlement processing will be skipped if the bank draft is issue via settlement. @param transactionContext @param isSettlementSrc @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				featureDTO @param bankDraftDTO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	The service will call the checkAndLevyFees() of FeeManagementService.  <li>It will be calculate and apply fees as per the price feature DTO.  <li>It will calculate the fee and also debit the fee from correspondent mode of settlement. <li>If the bank draft is liquidated(if liquidation mode =PAYMENT) then fee will not be derived @param sessionContext @param eventId @param featureDTO @param bankDraftDTO @return PriceFeatureDTO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	This method will calculate the fee for update status only. It will also calculate the net transaction amount and total fee amount based on the fee retrieved. @param transactionContext @param newBankDraft @param eventId @param featureDTO @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	determineFee	<p>This method is called to charge fee, if any applicable on Inward returned and Outward refused transaction. </p> @param accountNo @param currencyCode @param transactionReferenceNo @param subSequenceNo @param postingDate @param eventId @param payments @param PricingFactDTO [] facts	This method will determine the facts that will used to determine the fees applicable for this transaction. It will then calculate the fees using the facts and apply the fees over and above the transaction amount. @param transactionLog Indicates the transaciton log from where the fee facts will be derived by the engine @param messageLogItem TODO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchFee	<p>This method will be called while performing immediate credit transaction. The wrapped price feature dto will come as an input from the UI and fee management service will be called to perform the fee charge action as indicated in the wrapped price feature dto. </p> @param partyId @param currencies @param taskModuleType @param wrappedPriceFeatureDTO @param feeDetail @throws FatalException	This method is use dto determine and apply fee for immediate credit. @fcb.param context @fcb.param transactionLog @fcb.param creditRequestDTO @fcb.param eventId @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchFee	<p>This method will be called while performing immediate credit transaction. The wrapped price feature dto will come as an input from the UI and fee management service will be called to perform the fee charge action as indicated in the wrapped price feature dto. </p> @param partyId @param currencies @param taskModuleType @param wrappedPriceFeatureDTO @param feeDetail @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchFee	<p>This method will be called while performing immediate credit transaction. The wrapped price feature dto will come as an input from the UI and fee management service will be called to perform the fee charge action as indicated in the wrapped price feature dto. </p> @param partyId @param currencies @param taskModuleType @param wrappedPriceFeatureDTO @param feeDetail @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	This method logs the data into inquiry log table @fcb.param transactionLog @fcb.param messageLogItem @fcb.param insert
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	@see com.ofss.fc.app.pc.core.IOutgoingClearingService#doImmediateCreditByRef(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.dto.ImmediateCreditRequestDTO)
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	This method will insert details of Ledger in the table.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	addTransactionLogDetails	@param transactionLogDTO @param messageLogItemDTO @throws FatalException	This method will log data for TransactionLogDetail @param transactionLog @param messageLogItem @param insert
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	logTransactionCounter	<p>This method will log the number of transactions executed for an event for which fee is to be charged periodically. On the actual charge date, this count will be sent to pricing as an additional fact and based on this the amount to be charged will be determined. </p> @param accountId @param accountDomainCategory @param eventId @param asOnDate @param count @param transactionAmount @throws FatalException	This method will log the number of transactions processed from the provider account.     Every time a business transfer is initiated from CASA account, the account module maintains a count of the number of times such event was executed during the month. On month end, as part of periodic fee batch process, this action will be picked for execution. The account module will call pricing to inquire the amount to be charged as part of this process. Pricing will make a call back to CASA module to inquire the number of transactions executed for this event and return the net amount chargeable, which is stored tier based. Based on the amount returned, CASA will process the periodic fee charging.    <i>NOTE: On account opening itself the dates for charging periodic fee for business transfer transactions are sent by pricing and based on the date, CASA logs an action.</i>  @fcb.param MI, BulkTransferInstruction, bulkTransferInstruction @fcb.param MI, AccountingEntryType, accountingEntryType <li> {@link com.ofss.fc.enumeration.AccountingEntryType}</li> @fcb.param MI, String, eventId @fcb.param MI, Date, postingDate <li> {@link com.ofss.fc.datatype.Date}</li> @fcb.param MI,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				String, transactionReferenceNo @throws BusinessException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	logTransactionCounter	<p>&lt;p&gt;This method will log the number of transactions executed for an event for which fee is to be charged periodically. On the actual charge date, this count will be sent to pricing as an additional fact and based on this the amount to be charged will be determined. &lt;/p&gt;</p> <p>@param accountId @param accountDomainCategory @param eventId @param asOnDate @param count @param transactionAmount @throws FatalException</p>	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	logTransactionCounter	<p>&lt;p&gt;This method will log the number of transactions executed for an event for which fee is to be charged periodically. On the actual charge date, this count will be sent to pricing as an additional fact and based on this the amount to be charged will be determined. &lt;/p&gt;</p> <p>@param accountId @param accountDomainCategory @param eventId @param asOnDate @param count @param transactionAmount @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	This service validate the policy of immediate credit as well as perform the Immediate credit on provided data. It will raise an accounting event for the credit and also audit the transaction @throws FatalException @see com.ofss.fc.app.pc.core.IOutgoingCollectionService#doImmediateCredit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.dto.log.InstrumentTransferTransactionLogDTO, com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO, com.ofss.fc.datatype.Money)
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	This method is use dto determine and apply fee for immediate credit. @fcb.param context @fcb.param transactionLog @fcb.param creditRequestDTO @fcb.param eventId @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	This method will read the wrapped price feature dto provided in the input and perform fee processing for the tc encashment transaction. @param transactionDate @param componentPreferenceDTO @param wrappedPriceFeatureDTO @param payOutAmount @param eventId @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	This method will do the processing for returning the inward items which have been rejected. The return processing will involve the marking of the transaction state as CLOSED and creating new transaction with message state as RETURNED and authorisation state is AUTHORIZED and subsequent accounting between un-posted suspense and end-point. It will also log the transaction in scan reject queue. @param sessionContext @param type @param collectionType @param transactionReferenceNo @param transactionLog @param originalMessageLogItem @return @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	This method is used for determining the fee and levy fee for returns processing. @param inputTransactionLog @param instrumentType @param copiedTxnLog @param eventId @return @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	The service will call the checkAndLevyFees() of FeeManagementService.  <li>It will be calculate and apply fees as per the price feature DTO.  <li>It will calculate the fee and also debit the fee from correspondent mode of settlement. <li>If the bank draft is liquidated(if liquidation mode =PAYMENT) then fee will not be derived @param sessionContext @param eventId @param featureDTO @param bankDraftDTO @return PriceFeatureDTO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	@param transactionLog @param messageLogItem @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFee	@param feesChargingRequest @throws FatalException	
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		This method will read the wrapped price feature dto provided in the input and perform fee processing for the tc encashment transaction. @param transactionDate @param componentPreferenceDTO @param wrappedPriceFeatureDTO @param payOutAmount @param eventId @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		This method is used for determining the fee and levy fee for returns processing. @param inputTransactionLog @param instrumentType @param copiedTxnLog @param eventId @return @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		The service will call the checkAndLevyFees() of FeeManagementService.  <li>It will be calculate and apply fees as per the price feature DTO.  <li>It will calculate the fee and also debit the fee from correspondent mode of settlement. <li>If the bank draft is liquidated(if liquidation mode =PAYMENT) then fee will not be derived @param sessionContext @param eventId @param featureDTO @param bankDraftDTO @return PriceFeatureDTO
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	applyFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	reverseFees		@see com.ofss.fc.app.pc.billpay.service.biller.IBillPaymentService#reverseBillPayment(com.ofss.fc.app.context.SessionContext, com.ofss.fc.enumeration.dda.MessageType)
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	reverseFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	reverseFees		@param sessionContext @param bankDraft @param pricingAdapter @param eventId @param transactionType @throws FatalException
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	reverseFees		
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	reverseFees		<p>This method will reverse the fee charged for a transaction. </p> @param accountId @param accountModule @param eventId @param serviceId @param narrative @param atmNarration @param txnLiteral @param txnDescription
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchOperatingInstructionsForAccount		This method is used to populate the response message in case of successful transaction, for that it is using the input requestDTO and response for the each case. Inside the method it checks the type of transaction and based on that it populates use case based outputs. @param sessionContext @param fundTransferRequestDTO @param fundTransferResponse @param identifierType @param transactionAmount @param initiatorAccountId
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchOperatingInstructionsForAccount		The following method is used to fetch OperatingInstructionDetailsDTO given account number as input @param sessionContext @param accountNumber @param amount @return
IPaymentsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsAccountAdapter	fetchOperatingInstructionsForAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method is used to process the closure of the facility. In case there is no nominated account linked, then the fees will be fetched and will be marked as deferred and the caller will be provided with this info. In case the fees is deferred the expectation is that the facility will not be closed. TODO : 2.3 Dev Changes WIP @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method is used to check if the account being closed has been nominated for the facility fee charging. In case the account is a nominated account to any of the facility , the attribute isAccountLinked of {@link FacilityFeeInquiryResponse} will be set to true indicating the nomination. The module based on this attribute can provide the user with the option of proceeding with the transaction or not. In case the account being closed is linked to a non revolving facility, the system needs to check whether the fees needs to be collected during the transaction. This is because loan accounts might get closed and hence the fees might go into deferred state during the auto closure of non revolving facility (based on system parameter). To ensure that the pro rate fees are collected this activity is done. In case of back dated closure also the fees upto the current date needs to be collected and charged/deferred based on the usecase. Once this is done the facility will be marked for closure to ensure that no fees gets collected for the facility again during its closure or upto its closure. @param accountId, The account which is being closed. @param closureDate , The date on which the account is being closed. It can current or back date. @param acctCurrency, The currency of the account. @since 2.3 @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	@param appContext @param creditFacility @param computationDate @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method is used to check whether the facility has any deferred fee. In case of deferred fees, the facility cannot be closed.
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This initiates a dialog with pricing module and returns the pricing ticket number @fcb.param MI.com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @return {@link com.ofss.fc.app.lcm.dto.limits.facility.FacilityFeeInquiryResponses} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	generateTicket	This method generates a pricing ticket by calling the TicketService. This begins the start of the pricing interaction. The ticket number returned must be utilized for subsequent pricing calls within the same dialog. @param sessionContext @return PricingTicketDTO @throws FatalException	This method calls pricing to stamp the initial recommendations on a given ticket number. @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return EvaluatedPriceRecommendationDTO @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPriceOffering	This method calls pricing to stamp the initial recommendations on a given ticket number. @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return EvaluatedPriceRecommendationDTO @throws FatalException	This initiates the interaction with pricing module. This facilitates the creation of pricing recommendations on the facility. The initial recommendations are stored and linked in the following way: <ul> <li>The parent facility is identified from the input Credit facility structure.</li> <li>The fact information required is populated.</li> <li>The pricing module is called to resolve the pricing plan based on the facts and stamp the initial recommendations against the pricing ticket number.</li> <li>The pricing ticket number returned by the pricing module is stored against the parent facility fee placeholder. The pricing ticket number will be used for future transactions.</li> </ul> The pricing ticket number is generated and the initial pricing recommendations are stamped against the ticket number. This ticket number is then persisted into the database. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,CreditFacilityDTO[],facilityDTOArray, , This parameter captures the facility structure. The mainline or the parent facility id is derived and

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				the pricing ticket number is stored against the parent facility. @return {@link com.ofss.fc.app.lcm.dto.limits.facility.FacilityFeeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPriceOffering	This method calls pricing to stamp the initial recommendations on a given ticket number. @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return EvaluatedPriceRecommendationDTO @throws FatalException	This method calls pricing to stamp the initial recommendations on a given ticket number. @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return EvaluatedPriceRecommendationDTO @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	createPricingPlan	This will call pricing to create the pricing plan based on the recommendations stamped against the pricing ticket. The pricing plan will be created under the facility id, which will be treated as the arrangement id by the pricing module. @param pricingContractContextDTO @param facilityId @param evaluationDate @return PeriodicEventTrackerResponse @throws FatalException	This creates a pricing plan on the given facility id. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,com.ofss.fc.app.lcm.dto.limits.facility.creditfacility.CreditFacilityDTO,parentFacilityDTO, ,The mainline or the parent facility DTO.</br> @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	createPricingPlan	This will call pricing to create the pricing plan based on the recommendations stamped against the pricing ticket. The pricing plan will be created under the facility id, which will be treated as the arrangement id by the pricing module. @param pricingContractContextDTO @param facilityId @param evaluationDate @return PeriodicEventTrackerResponse @throws FatalException	This will call pricing to create the pricing plan based on the recommendations stamped against the pricing ticket. The pricing plan will be created under the facility id, which will be treated as the arrangement id by the pricing module. @param pricingContractContextDTO @param facilityId @param evaluationDate @return PeriodicEventTrackerResponse @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPeriodicPricesWithPeriodicDates	@param context @param arrangementNumber @param evaluationDate @param computationDate @param accountCurrency @param facts @param pricingTicketDTO @return PeriodicPriceFeatureDTO[] @throws FatalException	@param appContext @param creditFacility @param computationDate @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPeriodicPricesWithPeriodicDates	@param context @param arrangementNumber @param evaluationDate @param computationDate @param accountCurrency @param facts @param pricingTicketDTO @return PeriodicPriceFeatureDTO[] @throws FatalException	@param context @param arrangementNumber @param evaluationDate @param computationDate @param accountCurrency @param facts @param pricingTicketDTO @return PeriodicPriceFeatureDTO[] @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	computePeriodicFeesOnChargeDate	@param context @param arrangementNumber @param currencyCodes @param chargingDate @return @throws FatalException	@param applicationContext @param arrangementId @param chargingDate @param currencyCodes @param eventId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPriceForFacilityClosure	This method is used to get the price for facility closure. @param facilityId @param acctCurrency @param evaluationDate @param closeOrExpiryDate @param factArray @param currencyCodes @param pricingTicketDTO @return	This method is used to fetch the fees for the facility closure. The price details obtained from the pricing is returned as the output for further processing in the caller. @param facilityId @param currencyCode @param currencyCodeArray @param evaluationDate @param closeOrExpiryDate @param pricingTicketDTO @return WrappedPriceFeatureDTO @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPriceForFacilityClosure	This method is used to get the price for facility closure. @param facilityId @param acctCurrency @param evaluationDate @param closeOrExpiryDate @param factArray @param currencyCodes @param pricingTicketDTO @return	This method is used to check if the account being closed has been nominated for the facility fee charging. In case the account is a nominated account to any of the facility , the attribute isAccountLinked of {@link FacilityFeeInquiryResponse} will be set to true indicating the nomination. The module based on this attribute can provide the user with the option of proceeding with the transaction or not. In case the account being closed is linked to a non revolving facility, the system needs to check whether the fees needs to be collected during the transaction. This is because loan accounts might get closed and hence the fees might go into deferred state during the auto closure of non revolving facility (based on system parameter). To ensure that the pro rate fees are collected this activity is done. In case of back dated closure also the fees upto the current date needs to be collected and charged/deferred based on the usecase. Once this is done the facility will be marked for closure to ensure that no fees gets collected for the facility again during its closure or upto its closure. @param accountId, The account which is being closed. @param closureDate , The date on which the account is being closed. It can current or back date. @param acctCurrency, The currency of the account. @since 2.3 @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	getPriceForFacilityClosure	This method is used to get the price for facility closure. @param facilityId @param acctCurrency @param evaluationDate @param closeOrExpiryDate @param factArray @param currencyCodes @param pricingTicketDTO @return	This method is used to get the price if the facility is being closed or is expiring. @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	fetchDeferredFeesForFacility	This method is used to fetch the deferred fees for the facility id. @param facilityId @param currencyCode @param pricingTicketDTO @return	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	fetchDeferredFeesForFacility	This method is used to fetch the deferred fees for the facility id. @param facilityId @param currencyCode @param pricingTicketDTO @return	This method is used to check whether the facility has any deferred fee. In case of deferred fees, the facility cannot be closed.
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	fetchDeferredFeesForFacility	This method is used to fetch the deferred fees for the facility id. @param facilityId @param currencyCode @param pricingTicketDTO @return	This method is used to get the details of the deferred fees @throws FatalException @ accountId
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	markFeesAsDeferred	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param pricingEventId @throws FatalException	This method is used to process the closure of the facility. In case there is no nominated account linked, then the fees will be fetched and will be marked as deferred and the caller will be provided with this info. In case the fees is deferred the expectation is that the facility will not be closed. TODO : 2.3 Dev Changes WIP @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	markFeesAsDeferred	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param pricingEventId @throws FatalException	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	markFeesAsDeferred	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param pricingEventId @throws FatalException	@param appContext @param creditFacility @param computationDate @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	markFeesAsDeferred	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param pricingEventId @throws FatalException	
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	markFeesAsDeferred	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param pricingEventId @throws FatalException	This method is used to defer the fee in case the first disbursement has not occurred. @param applicationContext @param pricingTicketNumber @param eventId @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	closePricingPlan	This method is used to close the pricing plan when the facility expires or is closed. @param facilityId @throws FatalException	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	closePricingPlan	This method is used to close the pricing plan when the facility expires or is closed. @param facilityId @throws FatalException	This method is used to close the pricing plan when the facility expires or is closed. @param facilityId @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	fetchFeesForBackDatedClosure	@param facilityId @param acctCurrency @param evaluationDate @param closureDate @param factArray @param currencyCodes @param pricingTicketDTO @since 2.3 @return	This method is used to check if the account being closed has been nominated for the facility fee charging. In case the account is a nominated account to any of the facility , the attribute isAccountLinked of {@link FacilityFeeInquiryResponse} will be set to true indicating the nomination. The module based on this attribute can provide the user with the option of proceeding with the transaction or not. In case the account being closed is linked to a non revolving facility, the system needs to check whether the fees needs to be collected during the transaction. This is because loan accounts might get closed and hence the fees might go into deferred state during the auto closure of non revolving facility (based on system parameter). To ensure that the pro rate fees are collected this activity is done. In case of back dated closure also the fees upto the current date needs to be collected and charged/deferred based on the usecase. Once this is done the facility will be marked for closure to ensure that no fees gets collected for the facility again during its closure or upto its closure. @param accountId, The account which is being closed. @param closureDate , The date on which the account is being closed. It can current or back date. @param acctCurrency, The currency of the account. @since 2.3 @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	notifyOnFacilityUpdate	This method is used to notify the pricing regarding the updates in the facility. @param domainChangeSet @param effectiveDate @since 2.3 @throws FatalException	This method is used to check if there are changes in the sanctioned/limit amount or the approved term of the facility being updated. This information needs to be provided to the Pricing in case of any changes. @param facility @param facilityBeforeUpdate @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	intimatePricingOnClosureConfirmation	This method is used to notify pricing on the account closure confirmation. @param arrangementNumber @param accountCurrency @param evaluationDate @param factsArray @param pricesToBeDeferred @since 2.3 @throws FatalException	This method is used to do the activities when the closure of the account happens. @param appContext @param accountId @param closureDate @since 2.3 @throws FatalException
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	intimatePricing	This method is used to notify pricing when the deferred fees has been processed @param arrangementNumber @param accountCurrency @param evaluationDate @param factsArray @param pricesToBeDeferred @since 2.3 @throws FatalException	This method is used to do the activities when the closure of the account happens. @param appContext @param accountId @param closureDate @since 2.3 @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsPricingAdapter	intimatePricing	This method is used to notify pricing when the deferred fees  has been processed @param arrangementNumber @param accountCurrency @param evaluationDate @param factsArray @param pricesToBeDeferred @since 2.3 @throws FatalException	This processes the deferred fees for a facility. First it calls the settlement module  to process the deferred fees and then it will raise an accounting entry for the same. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,com.ofss.fc.app.lcm.dto.limits.facility.creditfacility.DeferredFeesAmountDetailsDTO[], deferredFeesAmountDetailsDTOArray, This contains the amount details required for the settlement process</br> @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	getSQLExpression	This method will SQL Expression for given filter ID. @param filterId The Filter id for which filter SQL needs to be derived	This method will SQL Expression for given filter ID. @param filterId The Filter id for which filter SQL needs to be derived
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	executeRule	This will execute the rule for the given rule Id and effective date.	
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	executeRule	This will execute the rule for the given rule Id and effective date.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	executeRuleSet	This will execute the rule set for the given rule set Id and effective date.	
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	getFactDetailsForRule	This method will list all the Facts and their respective dataTypes used while defining the Rule. @param ruleId Id of the Rule for which Facts needs to be retrieved @param effectiveDate Effective date of the Rule for which Facts needs to be retrieved @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside Rule @throws Exception If Error occurs while retrieving facts for rule	This method will list all the Facts and their respective dataTypes used while defining the Rule. @param ruleId Id of the Rule for which Facts needs to be retrieved @param effectiveDate Effective date of the Rule for which Facts needs to be retrieved. Date Format yyyy-MM-dd @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside Rule @throws Exception If Error occurs while retrieving facts for rule
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	getFactDetailsForRule	This method will list all the Facts and their respective dataTypes used while defining the Rule. @param ruleId Id of the Rule for which Facts needs to be retrieved @param effectiveDate Effective date of the Rule for which Facts needs to be retrieved @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside Rule @throws Exception If Error occurs while retrieving facts for rule	This method will list all the Facts and their respective dataTypes for a Domain And Category of ruleSetId. @param ruleSetId Id of the RuleSet for which Facts needs to be retrieved @param effectiveDate Effective date of the RuleSet for which Facts needs to be retrieved. Date Format yyyy-MM-dd @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside RuleSet @throws Exception If Error occurs while retrieving facts for ruleSet

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollectionRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollectionRulesAdapter	getFactDetailsForRuleSetDomainAndCategory	This method will list all the Facts and their respective dataTypes for a Domain And Category of ruleSetId. @param ruleSetId Id of the RuleSet for which Facts needs to be retrieved @param effectiveDate Effective date of the RuleSet for which Facts needs to be retrieved. @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside RuleSet @throws Exception If Error occurs while retrieving facts for ruleSet	This method will list all the Facts and their respective dataTypes for a Domain And Category of ruleSetId. @param ruleSetId Id of the RuleSet for which Facts needs to be retrieved @param effectiveDate Effective date of the RuleSet for which Facts needs to be retrieved. Date Format yyyy-MM-dd @return RuleFactDetailsInquiryResponse Containing array of FactDetails used inside RuleSet @throws Exception If Error occurs while retrieving facts for ruleSet
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	inquirePriceRecommendation	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationPricingAdapter	inquirePriceRecommendation	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException	
IIIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationPricingAdapter	inquirePriceRecommendation	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	inquirePriceRecommendation	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException	
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	inquirePriceRecommendation	This method will return the EvaluatedPriceRecommendationDTO i.e the recommendation for the price based on the internalId of the PricingTicket @param sessionContext :the session context @param pricingContractContextDTO : the pricing contract context dto @param asOnDate : transaction evaluation date @param currencyCodes : currency codes @return WrappedPriceFeatureInquiryResponse @throws FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPricingAdapter	getFeesForTicketNumber	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it. The arrangement number is a unique identifier used by modules to identify their respective entities. For eg: CASA, TD, Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number. This method will return all the price feature DTOS for a given ticket number @param currencyCodes @param ticketNumber @param sessionContext @return @throws FatalException	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it. The arrangement number is a unique identifier used by modules to identify their respective entities. For eg: CASA, TD, Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number. This method will return all the price feature DTOS for a given ticket number @param currencyCodes @param ticketNumber @param sessionContext @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPricingAdapter	getFeesForTicketNumber	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it. The arrangement number is a unique identifier used by modules to identify their respective entities. For eg: CASA, TD, Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number. This method will return all the price feature DTOS for a given ticket number @param currencyCodes @param ticketNumber @param sessionContext @return @throws FatalException	This method is used to populate fee for product Group of Offer Letter. @param context @param loanProductPriceGroup @return PriceDefinition @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPricingAdapter	getFeesForTicketNumber	<p>Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it. The arrangement number is a unique identifier used by modules to identify their respective entities. For eg: CASA, TD, Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number. This method will return all the price feature DTOS for a given ticket number</p> <p>@param currencyCodes @param ticketNumber @param sessionContext @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	getPricesWithLenientComputation	<p>Returns an array of type {@link WrappedPriceFeatureDTO}, containing the prices supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur &lt;p&gt; A {@link PricingTicket} is not required for this call. &lt;/p&gt; @param sessionContext holds the information of the party/customer @param pricingContextDTO the pricing context dto holds the info for pricing @param arrangementNumber the arrangement/account number of the party @param asOnDate the evaluation date @param currencyCodes a string array of the currency codes @return an array of type {@link WrappedPriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception</p>	<p>Returns an array of type {@link WrappedPriceFeatureDTO}, containing the prices supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur &lt;p&gt; A {@link PricingTicket} is not required for this call. &lt;/p&gt; @param sessionContext holds the information of the party/customer @param pricingContextDTO the pricing context dto holds the info for pricing @param arrangementNumber the arrangement/account number of the party @param asOnDate the evaluation date @param currencyCodes a string array of the currency codes @return an array of type {@link WrappedPriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPricingAdapter	getPricesWithLenientComputation	<p>Returns an array of type <a href="#">{@link WrappedPriceFeatureDTO}</a>, containing the prices supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur</p> <p>&lt;p&gt; A <a href="#">{@link PricingTicket}</a> is not required for this call. &lt;/p&gt;</p> <p>@param sessionContext holds the information of the party/customer  @param pricingContextDTO the pricing context dto holds the info for pricing  @param arrangementNumber the arrangement/account number of the party  @param asOnDate the evaluation date  @param currencyCodes a string array of the currency codes  @return an array of type <a href="#">{@link WrappedPriceFeatureDTO}</a>  @throws FatalException if any of the arguments are invalid or if the <a href="#">{@link EventPriceService}</a> throws an exception</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	fetchRatesForRecommendation	This service is to provide the net rate on recommendation for simulation purposes. Send all the facts required in {@link PricingContractContextDTO} to evaluate the rate. This service works on a existing recommendation. So before calling this create and recommendation. And send the pricing ticket number to this service. @param sessionContext @param pricingContextDTO @param stageDetails @param asOnDate @param dateToFilterOn @return @throws FatalException	This service is to provide the net rate on recommendation for simulation purposes. Send all the facts required in {@link PricingContractContextDTO} to evaluate the rate. This service works on a existing recommendation. So before calling this create and recommendation. And send the pricing ticket number to this service. @param sessionContext @param pricingContextDTO @param stageDetails @param asOnDate @param dateToFilterOn @return @throws FatalException
IntegrationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPricingAdapter	fetchRatesForRecommendation	This service is to provide the net rate on recommendation for simulation purposes. Send all the facts required in {@link PricingContractContextDTO} to evaluate the rate. This service works on a existing recommendation. So before calling this create and recommendation. And send the pricing ticket number to this service. @param sessionContext @param pricingContextDTO @param stageDetails @param asOnDate @param dateToFilterOn @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireAccountScheduleDefinition	This method is used to inquire about Account Schedule Information. Along with the account schedule,  it also returns additional information about the account schedule @return {@link LoanScheduleInquiryResponse} This class holds the loan account schedule stages and more @throws com.ofss.fc.infra.exception.FatalException	This method is used to inquire about Account Schedule Information. Along with the account schedule,  it also returns additional information about the account schedule @return {@link LoanScheduleInquiryResponse} This class holds the loan account schedule stages and more @throws com.ofss.fc.infra.exception.FatalException
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireAccountScheduleDefinition	This method is used to inquire about Account Schedule Information. Along with the account schedule,  it also returns additional information about the account schedule @return {@link LoanScheduleInquiryResponse} This class holds the loan account schedule stages and more @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. @return {@link LoanAccountConfigurationResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. @return {@link LoanAccountConfigurationResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. @return {@link LoanAccountConfigurationResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. @return {@link LoanAccountConfigurationResponse} @throws com.ofss.fc.infra.exception.FatalException
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	inquireLoanAccountConfiguration	This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id. @return {@link LoanAccountConfigurationResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IIntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLoanAdapter	fetchRateAndRepaymentDetailsForDocumentGeneration	<p>This method is responsible for providing the details required for Document Generation.   It uses Loan Schedule Configuration and Loan Account Configuration to derive the Rate details and Repayment details for every rate details. </p> @param sessionContext @param arrangementId @throws FatalException	<p>This method is responsible for providing the details required for Document Generation.   It uses Loan Schedule Configuration and Loan Account Configuration to derive the Rate details and Repayment details for every rate details. </p> @param sessionContext @param arrangementId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationLoanAdapter	fetchRateAndRepaymentDetailsForDocumentGeneration	<p>This method is responsible for providing the details required for Document Generation.   It uses Loan Schedule Configuration and Loan Account Configuration to derive the Rate details and Repayment details for every rate details. </p> @param sessionContext @param arrangementId @throws FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IntegrationLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationLoanAdapter	fetchMostRecentValidRateLockConfigurationForAccountId	This method returns the most recent & valid RateLock for the given arrangementId. If there is no valid RateLock, then, a null RateLockConfigurationDTO is returned. @param sessionContext @param accountId @param effectiveDate @return @throws FatalException	This method returns the most recent & valid RateLock for the given arrangementId. If there is no valid RateLock, then, a null RateLockConfigurationDTO is returned. @param sessionContext @param accountId @param effectiveDate @return @throws FatalException
CommonServiceTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceTDAdapter	checkIfAllIDepositsAreClosed		This is the helper method to check whether a particular BranchParameters record can be closed or not.  If the BranchParameters can be closed, it returns true, else it returns false. This method checks whether any account(s) is (are) opened under any domain category TD, CASA, Loan,  If no account is opened , the branch can be closed, otherwise, we should not allow to close the branch. @param sessionContext : {@link SessionContext} @param branchCode : String @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	validatePartyDetails	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	validatePartyDetails	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	validatePartyDetails	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	<p>This method is used to check party . @param sessionContext @param partyId @param errorCode @param className @param attributeName</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	validatePartyDetails	This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException	
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	fetchProductCode	This method will return the product code for the given account and its type @param sessionContext @param accountCode @param accountType @return @throws FatalException	
ISettlementPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPartyAdapter	fetchProductCode	This method will return the product code for the given account and its type @param sessionContext @param accountCode @param accountType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILinkedAccountInfoFromLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ILinkedAccountInfoFromLoansAdapter	inquireAccountInformation		
IOriginationTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAAdapter	updateTdAccountPreferences	<p>This method is used to call the update Account Configurations service in the Term Deposit Module.It updates &lt;br&gt; the Account preferences for an existing TermDeposit Account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	<p>This method is used to call the update Account Configurations service in the Term Deposit Module.It updates &lt;br&gt; the Account preferences for an existing TermDeposit Account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	updateTdAccountPreferences	This method is used to call the update Account Configurations service in the Term Deposit Module.It updates   the Account preferences for an existing TermDeposit Account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @throws FatalException	This method is used to update the Account Configurations for a Term Deposit Account that has already been opened previously. It updates the following informations inside the Account Configuration.  1. Payout instructions provided for the account.  2. Financial and non financial Operating instructions for the account.  3. Default communication address for the account are updated.  @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This method inserts {@link OperatingInstructionDTO} for the Account from the given {@link TDAccountConfigurationDTO} object. @param sessionContext @param tdAccountConfigurationDTO @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This method is used to validate the Account before assigning an Operating Instruction to it. If the account is not valid one then it will throw {@link RecordNotFoundException} @param sessionContext @param accountId @param accountType @return @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Other.&lt;br&gt; For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.&lt;br&gt; Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product  Refactored on March 27 to change the input from productGroupLinkageDTO to ContractDetailComputationRequest. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ContractDetailComputationRequest, contractDetailComputationRequest, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateAccount	This method will validate the account whether exists or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, accountId,, Indicates the identifier for the account @throws FatalException	This method will insert/update the account operating instructions for account when secondary applicant is added to the account from origination. @param sessionContext @param accountId @param holderPreferenceDTO @param operatingInstructionCategoryType @param individualPartyRole @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateProductDetails	<p>This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product</p> <p><code>@fcb.param MI,SessionContext,sessionContext, ,Session Context</code> which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><code>@fcb.param MI, TermDepositProductGroupLinkage DTO, productGroupLinkageDTO, ,</code> Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s)</p> <p><code>@return {@link com.ofss.fc.service.response.TransactionStatus}</code> An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.</p> <p><code>@throws FatalException</code></p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateProductDetails	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateProductDetails	<p>This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkage DTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	validateProductDetails	<p>This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	<p>This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getNetInterestRate	This service will fetch the net interest rate for the account for the facts passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder. @fcb.param Map<String, Object>,tdFactsDTO, , Indicates the facts passed for the term deposit account such as requested amount , term and interest payout frequency @return BigDecimal, , Indicates the net interest rate applicable for the term deposit @throws FatalException	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getNetInterestRate	This service will fetch the net interest rate for the account for the facts passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,td	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration.Separate calls for fetching interest rates, margin rates and servicing charges are then done to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration. This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder.&lt;br&gt;  @fcb.param Map&lt;String, Object&gt;,tdFactsDTO, , Indicates the facts passed for the term deposit account such as requested amount , term and interest payout frequency  @return BigDecimal, , Indicates the net interest rate applicable for the term deposit @throws  FatalException</p>	<p>the pricing engine using the same pricing ticket number. @fcb.param  MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param  MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration  @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application  @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return  NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit.  @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getNetInterestRate	<p>This service will fetch the net interest rate for the account for the facts passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder.&lt;br&gt;@fcb.param Map&lt;String, Object&gt;,tdFactsDTO, , Indicates the facts passed for the term deposit account such as requested amount , term and interest payout frequency @return BigDecimal, , Indicates the net interest rate applicable for the term deposit @throws FatalException</p>	<p>This API calculates the net interest rate , for TD, for the given account configuration. @param sessionContext @param submissionId @param productGroupSerialNumber @param simulationID @return NetInterestRateResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getNetInterestRate	<p>This service will fetch the net interest rate for the account for the facts passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder.&lt;br&gt;@fcb.param Map&lt;String, Object&gt;,tdFactsDTO, , Indicates the facts passed for the term deposit account such as requested amount , term and interest payout frequency @return BigDecimal, , Indicates the net interest rate applicable for the term deposit @throws FatalException</p>	<p>This method returns the net interest rate based on the information present in {@link TDAccountConfigurationDTO}. It populates the TD facts and pass it to the service getNetRateAtRecommendation in the TD PricingAdapter. This service returns the net interest rate. @param sessionContext @param tdAccountConfigurationDTO @param tdFactsDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getNetInterestRate	This service will fetch the net interest rate for the account for the facts passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration.This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder. @fcb.param Map<String, Object>,tdFactsDTO, , Indicates the facts passed for the term deposit account such as requested amount , term and interest payout frequency @return BigDecimal, , Indicates the net interest rate applicable for the term deposit @throws FatalException	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module.  This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	<p>This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level.</p> <p>@fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening.</p> <p>@fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.	This service will populate pricing facts specific to Term Deposit products. @param sessionContext @param tdAccountConfigurationDTO @param productCode @param tdFactsDTO
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String,	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>offerId, , Indicates the identifier for the offer used for account opening.  @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	<p>such as user id, branch, branch code, channel etc.  @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	<p>This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level.</p> <p>@fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening.</p> <p>@fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	<p>This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level.</p> <p>@fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening.</p> <p>@fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	<p>This method uses requested tenor and interest start date to find out the Term deposit maturity date . It populates the term deposit maturity date in the form of {@link Date} @param requestedTenor @param interestStartDate @return maturityDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	<p>This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level.</p> <p>@fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	<p>This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product&lt;br&gt; Refactored on March 27 to change the input from productGroupLinkageDTO to ContractDetailComputationRequest. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ContractDetailComputationRequest, contractDetailComputationRequest, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	<p>This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level.</p> <p>@fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	<p>This method uses interest start date and maturity date to find out the Term deposit tenor. It populates the term deposit tenor in the form of {@link DurationDTO} @param interestStartDate @param offerId @param maturityDate @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.	This method returns the net interest rate based on the information present in {@link TDAccountConfigurationDTO}. It populates the TD facts and pass it to the service getNetRateAtRecommendation in the TD PricingAdapter. This service returns the net interest rate. @param sessionContext @param tdAccountConfigurationDTO @param tdFactsDTO @return @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String,	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>offerId, , Indicates the identifier for the offer used for account opening.  @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.</p>	<p>27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositTenor	This service will calculate the term deposit term or tenor for the maturity date passed. It will verify the term parameter set for the product i.e. if the term applicable is in years, months or days or any combination of the above and return the tenor in the same format as specified at product level. @fcb.param MI, Date, interestStartDate, , Indicates the interest start date for the term deposit @fcb.param MI, String, offerId, , Indicates the identifier for the offer used for account opening. @fcb.param MI, Date, maturityDate, , Indicates the date on which the term deposit will mature. @return DurationDTO, , Returns the requested term in form of years, months and days for the maturity date specified.	This service will populate pricing facts specific to Term Deposit products. @param sessionContext @param tdAccountConfigurationDTO @param productCode @param tdFactsDTO
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened.  @throws	This method is used to call the DDA module to validate the Account Id for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will fetch the payout instructions for the first deposit on the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method is used to call the DDA module to validate the Account Id for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will fetch the payout instructions for the first deposit on the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened.  @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method checks whether an TD account is valid or not. If the account is not valid one, then it will throw an exception with proper error message. @param sessionContext @param accountId @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	<p>This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened.&lt;br&gt;@throws FatalException</p>	<p>This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.&lt;br&gt; If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.&lt;br&gt; For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.&lt;br&gt; Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IOrigination TDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method is used to log event related to Account creation of demand deposit. @param sessionContext @param submissionId @param applicationId @param response @param prodGrpLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This method is used to log event related to Account creation of term deposit. @param sessionContext @param response @param submissionId @param applicationId @param prodGrpLinkage
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will populate the simulation details for the product group for a given Deposit application. It will iterate for all the product group linkages which are part of the application. For every Product offer references inside the product group linkage, it will fetch and stores the simulation if there is anything available. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DepositApplicationDTO, applicationDTO, , Indicates the deposit application for which simulation details are being populated. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will fetch account configurations done for a CASA account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g: Internet banking, Debit Card, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the account configuration details for demand deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will fetch account configurations done for a TD account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g: Internet banking, Phone banking, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the account configuration details for term deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchAccount	This method is used to call the TD module to get the account details for a Term Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will mark the input product group as withdrawn or rejected. If the whole submission has been withdrawn, all product groups within the submission will be withdrawn.  If a single product group is rejected (by rejecting any document at product group level) or application is rejected (by rejecting the offer letter), the product group state is marked as REJECTED.   Facility details are deleted in both the above cases  If the product group is a part of the bundle, a bundle exception log is generated and saved.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO productGroupLinkageDTO, , Indicates the product group being withdrawn or rejected. @fcb.param MI,String event,Indicates the corresponding event set by the caller @fcb.param CI,String nominatedOfferId, , Indicates the offer identifier for the product group nominated for bundle fees. This is applicable only if product group is part of a bundle @fcb.param CI,String applicationBranch, ,Indicates the application branch. This is applicable only if product group is part of a bundle @fcb.param OI,String withDrawComments, , Indicates the reason for withdrawal of product group. This is applicable only if product group is part of a bundle @fcb.param MI,boolean isMarkedForExpiry,Indicates the submission is marked for expiry @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPayoutInstructions	This service will fetch the payout instructions for the first deposit on the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Term Deposit Account already opened. @throws FatalException	This service will fetch account configurations done for a TD account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g: Internet banking, Phone banking, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the account configuration details for term deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openAccount	This method is used to call the openAccount method in the Term Deposit Module.It opens a TD Account and generates the Account Number. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account   configuration.This class contains information about the preferences given for the term deposit account  as well as preferences captured for each account holder.  @return TermDepositCreationResponse,This class represents the response Object that is sent after opening a new Term deposit Account and Deposits for the account. @throws FatalException	This method is used to call the openAccount method in the Term Deposit Module.It opens a TD Account and generates the Account Number. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account   configuration.This class contains information about the preferences given for the term deposit account  as well as preferences captured for each account holder.  @return TermDepositAccountResponse,This class represents the response Object that is sent after opening a new Term deposit Account and Deposits for the account. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openAccount	<p>This method is used to call the openAccount method in the Term Deposit Module. It opens a TD Account and generates the Account Number. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account configuration. This class contains information about the preferences given for the term deposit account as well as preferences captured for each account holder. @return TermDepositCreationResponse, This class represents the response Object that is sent after opening a new Term deposit Account and Deposits for the account. @throws FatalException</p>	<p>This service will create a term deposit account for given account configuration. It is wrapper service called from process. It does the following in sequence 1. Calls the TermDepositAccountApplicationService to - populate the TD account opening DTO - open a TD account The account id is set in the response. 2. Adds the account reference to the corresponding offer in the application in origination. 3. Performs state transition for the product group and the application, if necessary @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the identifier for the submission @fcb.param MI, String, applicationId, , Indicates the identifier for the application @fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application. @fcb.param MI, int, offerSerialNumber, , Indicates the offer serial number @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , This class holds account configuration details for the term deposit offer. It contains information about the accounts being configured within the simulation such as payin details, payout etc. @return {@link TermDepositAccountResponse} This class contains the account opening response i.e. account id @exception &lt;br&gt; {@link OriginationErrorConstants#OR_DEPOSIT_APPLICATION_INVALID} :FC_OR_ST_0002 &lt;br&gt; This exception occurs during invalid state transition where the application is not a Deposit application &lt;br&gt; &lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openTermDeposit		
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openTermDeposit		<p>&lt;br&gt;This method is used to open a TD deposit using the {@link TDAccountConfigurationDTO}. If the ImmediatePayinRequired field is true then call is made to {@link DepositApplicationService}.openDeposit to open the term deposit.&lt;br&gt; @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account &lt;br&gt; configuration.This class contains information about the preferences given for the term deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return PayinResponse,This class represents the response Object that is sent after opening a new Term deposit for the account. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openTermDeposit		<p>This service will create a term deposit account for given account configuration. It is wrapper service called from process. It does the following in sequence</p> <ol style="list-style-type: none"> <li>1. Calls the TermDepositAccountApplicationService to - populate the TD account opening DTO - open a TD account The account id is set in the response.</li> <li>2. Adds the account reference to the corresponding offer in the application in origination.</li> <li>3. Performs state transition for the product group and the application, if necessary</li> </ol> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p>@fcb.param MI,String, submissionId, , Indicates the identifier for the submission</p> <p>@fcb.param MI,String, applicationId, , Indicates the identifier for the application</p> <p>@fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application.</p> <p>@fcb.param MI, int, offerSerialNumber, , Indicates the offer serial number</p> <p>@fcb.param MI,TDAccountConfigurationDTO, tdAccountConfigurationDTO, , This class holds account configuration details for the term deposit offer. It contains information about the accounts being configured within the simulation such as payin details, payout etc.</p> <p>@return {@link TermDepositAccountResponse}</p> <p>This class contains the account opening response i.e. account id</p> <p>@exception &lt;br&gt; {@link OriginationErrorConstants#OR_DEPOSIT_APPLICATION_INVALID} :FC_OR_ST_0002 &lt;br&gt; This exception occurs during invalid state transition where the application is not a Deposit application &lt;br&gt; &lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	openTermDeposit		<p>&lt;br&gt;This method is used to open a TD deposit using the {@link TDAccountConfigurationDTO}. If the ImmediatePayinRequired field is true then call is made to {@link DepositApplicationService}.openDeposit to open the term deposit.&lt;br&gt; @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO,tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account &lt;br&gt; configuration.This class contains information about the preferences given for the term deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return PayinResponse,This class represents the response Object that is sent after opening a new Term deposit for the account. @throws FatalException</p>
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositMaturityDate	This method uses requested tenor and interest start date to find out the Term deposit maturity date . It populates the term deposit maturity date in the form of {@link Date} @param requestedTenor @param interestStartDate @return maturityDate	
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositMaturityDate	This method uses requested tenor and interest start date to find out the Term deposit maturity date . It populates the term deposit maturity date in the form of {@link Date} @param requestedTenor @param interestStartDate @return maturityDate	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration.Separate calls for fetching interest rates,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param  MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param  MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit.  @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositMaturityDate	This method uses requested tenor and interest start date to find out the Term deposit maturity date . It populates the term deposit maturity date in the form of {@link Date} @param requestedTenor @param interestStartDate @return maturityDate	
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	getTermDepositMaturityDate	This method uses requested tenor and interest start date to find out the Term deposit maturity date . It populates the term deposit maturity date in the form of {@link Date} @param requestedTenor @param interestStartDate @return maturityDate	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	logProposedPayin	<p>This method logs the request for proposed TD. This is done if customer only opens the TD account but &lt;br&gt; wants to fund the TD later. Also in the payin scenarios where payin fails due to insufficient funds in the &lt;br&gt; casa account, proposed TD is logged @param sessionContext @param proposedPayinDTO @return TransactionStatus @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	logProposedPayin	This method logs the request for proposed TD. This is done if customer only opens the TD account but   wants to fund the TD later. Also in the payin scenarios where payin fails due to insufficient funds in the   casa account, proposed TD is logged @param sessionContext @param proposedPayinDTO @return TransactionStatus @throws FatalException	This method is used to log the ProposedPayinDTO details for Term Deposit. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDProposedPayinDTO,proposedPayinDTO, ,This class has the minimal details captured for creating a proposed Term Deposit . This is processed when a payment is received towards the proposed term deposit. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	logProposedPayin	This method logs the request for proposed TD. This is done if customer only opens the TD account but   wants to fund the TD later. Also in the payin scenarios where payin fails due to insufficient funds in the   casa account, proposed TD is logged @param sessionContext @param proposedPayinDTO @return TransactionStatus @throws FatalException	This method is used to log the ProposedPayinDTO details for Term Deposit. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDProposedPayinDTO,proposedPayinDTO, ,This class has the minimal details captured for creating a proposed Term Deposit . This is processed when a payment is received towards the proposed term deposit. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAadapter	updateAccountHolderPreferences	This method updates the account holder preferences for joint applicant who was not present   at the time of account opening @param sessionContext @param accountId @param tdAccountHolderPreferenceDTO @param isUpdateAccountRelationshipRequired @return TransactionStatus @throws FatalException	
IOriginationTDAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAadapter	updateAccountHolderPreferences	This method updates the account holder preferences for joint applicant who was not present   at the time of account opening @param sessionContext @param accountId @param tdAccountHolderPreferenceDTO @param isUpdateAccountRelationshipRequired @return TransactionStatus @throws FatalException	This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relationship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	updateAccountHolderPreferences	<p>This method updates the account holder preferences for joint applicant who was not present &lt;br&gt; at the time of account opening  @paramsessionContext @param accountld @param tdAccountHolderPreferenceDTO @param isUpdateAccountRelationshipRequired @return TransactionStatus @throws FatalException</p>	<p>This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.&lt;br&gt; If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.&lt;br&gt; For accounts opened with 2 or more applicants, the existing party relationships remain</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.&lt;br&gt; Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationShipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This service is used for fetching Pricing Details based on TD Pricing Input DTO @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DDA Pricing Input DTO, ddaPricingInputDTO @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI, TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI, ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception   {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This service is used for fetching Pricing Details based on TD Pricing Input DTO @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAPricingInputDTO, ddaPricingInputDTO @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This API nominates an account for the facility to be nominated for fees. It performs the following on the submission <ol> <li>fetch LendingProductGroupLinkageDTO</li> <li>fetch OfferReferenceDTO</li> <li>update the isNominatedForFacilityFees flag to true</li> <li>regenerate pricingTicketNumber</li> <li>fetch CreditFacilityDTO</li> <li>create and set CreditFacilityFeeDTO into CreditFacilityDTO with updated flag for nominatedAccount which is simulationId if it is a new account else it is CreditFacilityFeeDTO.getNominatedAccount()</li></ol> @param sessionContext @param submissionId @param creditFacilityFeeDTO @param simulationId @return AccountResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This API calculates the net interest rate , for TD, for the given account configuration. @param sessionContext @param submissionId @param productGroupSerialNumber @param simulationID @return NetInterestRateResponse @throws FatalException
IOriginationTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationTDAdapter	fetchPricingDetails		This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  This service is refactored to this module from { <a href="#">@link</a> TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception   {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTermDepositsInformation		@return Object
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTermDepositsInformation		@param sessionContext @param partyId @return LoanAccountInformationResponse @throws FatalException p
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTermDepositAccounts		@param sessionContext @param partyId @return LoanAccountInformationResponse @throws FatalException p
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTermDepositAccounts		Service to fetch all regular CASA accounts, TD accounts and external accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTermDepositAccounts		Service to fetch all CASA accounts and TD accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	updateTermDepositAccountStatus		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	updateTermDepositAccountStatus		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	updateTermDepositAccountStatus		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchTDInformation		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	updateTDStatusToDebitBlock		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	blockTermDepositAccounts		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	blockTermDepositAccounts		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	modifyStatusOfTermDepositAccounts		
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	modifyStatusOfTermDepositAccounts		Method to update the status of all the accounts held by the customer to Regular, in case customer is removed as Blacklisted. @param partyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	modifyStatusOfTermDepositAccounts		Method to update the status of all the accounts held by the customer to Total Block, in case customer is Blacklisted. @param partyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	primaryAccountHolderIDMaintenance		Maintain primary Account Holder ID for TD Account. @param accountId,newpartyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the Loan module for the change in ownership.  @param accountId @param newPrimaryPartyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the TD module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the DDA module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchAllDeposits	Method to fetch TD balances for a given accountId @param accountId @param depositStatus @throws FatalException	Method to fetch TD balances for a given accountId @param accountId @param depositStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	fetchAllDeposits	Method to fetch TD balances for a given accountId @param accountId @param depositStatus @throws FatalException	<p>&lt;p&gt;This method takes a term deposit account number as input and fetches details of all the deposits under this account, ignoring the status of the deposit. &lt;br/&gt;&lt;i&gt;Account id is a mandatory input to this service.&lt;/i&gt; &lt;br/&gt; &lt;br/&gt;This service first validates the term deposit account and then goes ahead to inquire the account details.&lt;br/&gt; The response from this service includes the following details for each deposit under the term deposit account: &lt;ul&gt;&lt;li&gt;Deposit Id&lt;/li&gt; &lt;li&gt;Offer Id&lt;/li&gt; &lt;li&gt;Renewal Counter&lt;/li&gt; &lt;li&gt;Deposit Status - which could be any one of these Regular or Closed or Unclaimed or Matured or FundAwaited&lt;/li&gt; &lt;li&gt;Deposit Terms - including term, auto renewal term, force renewal term for Lien&lt;/li&gt; &lt;li&gt;Certificate Status - which could be any one of these NotIssued or Issued or Duplicate or FileWithBank or Lost or ReIssued&lt;/li&gt; &lt;li&gt;Indicator if Lien is allowed&lt;/li&gt; &lt;li&gt;Indicator if is Blocked&lt;/li&gt; &lt;li&gt;Balance - including principle balance, maturity amount, closeout balance, uncollected principle balance etc.&lt;/li&gt; &lt;li&gt;Payout details - including interest payout frequency, next interest payout date&lt;/li&gt; &lt;li&gt;DepositDates - including deposit date, maturity date, closure date etc.&lt;/li&gt; &lt;li&gt;Lien Amount and Total Lien Principle Amount&lt;/li&gt; &lt;li&gt;Net Interest Rate and Total Lien Interest Amount&lt;/li&gt; &lt;li&gt;User defined margin applicable&lt;/li&gt; &lt;/ul&gt; &lt;/p&gt; @param sessionContext @param accountId @return {@link com.ofss.fc.app.td.dto.deposit.DepositResponse} @throws FatalException</p>





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerTermDepositsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerTermDepositsAdapter	getOfferId		This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray
IHardshipProcessForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\IHardshipProcessForApprovalRequestAdapter	initiateHardshipProcessForApprovalProcess		@see com.ofss.fc.app.collection.service.hardshipprocessforapproval.IHardshipProcessForApprovalApplicationService This method creates case from HardshipProcessForApprovalRequestDTO @param p_SessionContext @param p_HardshipProcessForApprovalRequestDTO @throws FatalException
IHardshipProcessForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\IHardshipProcessForApprovalRequestAdapter	initiateHardshipProcessForApprovalProcess		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	This method is used to add a subline. It loads {@link IAccountLCMAAdapter} and invokes {@link IAccountLCMAAdapter#addFacility()} method. @param sessionContext @param facilityDTO @return transactionStatus @throws FatalException
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is used to update a previously maintained subline. It loads {@link IAccountLCMAAdapter} and invokes {@link IAccountLCMAAdapter#updateFacility()} method. @param sessionContext @param facilityDTO @return transactionStatus @throws FatalException
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	fetchFacilityDetails	@param sessionContext @param facilityId @return @throws FatalException	This method will fetch all the details related to facility Id.This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAadapter	fetchMainFacilityForAccount	To fetch the main facility for an account that is linked to a sub facility. @param sessionContext the {@link SessionContext} @param accountId the Account Id @throws FatalException	This method creates a new Account Variation Request for the given account code.  It first fetches the parent facility linked to the given account.  Using this facility, it creates a Submission Request. This submission request is linked to the account variation request and is persisted into the database. It returns the newly created account variation request and the serialized DTO @param sessionContext the {@link SessionContext} @param accountCode the Account Code @return the newly created account variation request and the serialized DTO @throws FatalException
IAccountLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAadapter	fetchMainFacilityForAccount	To fetch the main facility for an account that is linked to a sub facility. @param sessionContext the {@link SessionContext} @param accountId the Account Id @throws FatalException	@see IAccountLCMAadapter#fetchMainFacilityForAccount(SessionContext, String)
IAccountLCMAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAadapter	convertProposedFacilityToConfirmed	This service will 1. Activate or confirm New Proposed Facilities 2. Collateral Group Linkage confirm 3. Confirm the linkage of Covenants, Conditions to the facility All the above can be achieved by the method convertProposedToCreditFacility(facilityId) of ProposedFacilityApplicationService @param sessionContext the {@link SessionContext} @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException	This service will 1. Activate or confirm New Proposed Facilities 2. Collateral Group Linkage confirm 3. Confirm the linkage of Covenants, Conditions to the facility All the above can be achieved by the method convertProposedToCreditFacility(facilityId) of ProposedFacilityApplicationService @param sessionContext the {@link SessionContext} @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	convertProposedFacilityToConfirmed	This service will 1. Activate or confirm New Proposed Facilities 2. Collateral Group Linkage confirm 3. Confirm the linkage of Covenants, Conditions to the facility All the above can be achieved by the method convertProposedToCreditFacility(facilityId) of ProposedFacilityApplicationService @param sessionContext the {@link SessionContext} @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException	This method will confirm an Account Variation Request.  It will first confirm the proposed facilities for the variation request  It will then do the variation request transaction depending the variation request type. For eg., for {@link VariationRequestType#LOAN_ROLLOVER_REQUEST}, it will call the Loan module's {@link IAccountLoanAdapter#addLoanRolloverInstruction(SessionContext, com.ofss.fc.app.loan.dto.rolloverinstruction.LoanRolloverInstructionDTO, com.ofss.fc.app.loan.dto.rate.lock.RateLockDTO, com.ofss.fc.app.loan.dto.fee.collection.FeeCollectionDTO)} .  It will then mark the variation request status as completed. @param sessionContext the {@link SessionContext} @param variationRequestId the Variation Request Id @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException
IAccountLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLCMAAdapter	fetchGuarantorsForAccount	This method will take input string array of account number and sessionContext.It will call to FacilityInquiryApplicationService.fetchGuarantorsforFacilityandAccount(sessionContext, null, accountsArray).String array of facilityIds is not required.On the basis of account number it will return FacilityGuarantorsResponse. @param sessionContext the {@link SessionContext} @param accountIdsArray the accountIdsArray Id @return the {@link FacilityGuarantorsResponse} @throws FatalException	This method will fetch AccountGuarantors by calling fetchGuarantorsForAccount() from AccountLCMAAdapter.This method will accountGuarantorsDTO. @param sessionContext @param accountGuarantorsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDelayedTransactionExecutionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IDelayedTransactionExecutionAdapter	executeDelayedTransaction	This method will execute DelayedTransactionPoller @throws FatalException	This method will execute DelayedTransactionPoller @throws FatalException
IDelayedTransactionExecutionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IDelayedTransactionExecutionAdapter	executeDelayedTransaction	This method will execute DelayedTransactionPoller @throws FatalException	Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller, delayed transaction poller, FTP webservice poller. Please add more in this method. Additionally we also have an extension for consulting pollers @param sessionContext
IPopulateGrantedValuesWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPopulateGrantedValuesWorkflowAdapter	getForType		getForType
IPopulateGrantedValuesWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPopulateGrantedValuesWorkflowAdapter	getValueToBeChecked		getValueToBeChecked
IChannelDSLAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDSLAdapter	getBasicInformationDetailsDTO	The following method returns the {@link BasicInformationDetailsDTO} BasicInformationDetailsDTO, which contains information required for the limit validation. @param businessContextDTO @param conversationText @return BasicInformationDetailsDTO @throws FatalException	This method fetches {@link BasicInformationDetailsDTO} @param businessContextDTO @param conversationText @return BasicInformationDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDSL Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDSLAdapter	getBasicInformationDetailsDTO	The following method returns the {@link BasicInformationDetailsDTO} BasicInformationDetailsDTO, which contains information required for the limit validation. @param businessContextDTO @param conversationText @return BasicInformationDetailsDTO @throws FatalException	This method returns the instance of channel adapter. @return ChannelDSLAdapter class instance
IChannelDSL Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDSLAdapter	getBasicInformationDetailsDTO	The following method returns the {@link BasicInformationDetailsDTO} BasicInformationDetailsDTO, which contains information required for the limit validation. @param businessContextDTO @param conversationText @return BasicInformationDetailsDTO @throws FatalException	This methods fetches the transaction amount through an adapter call and performs the limit validation for channel EMAIL. @param sessionContext the session context to be used. @param channelContext the channel context to be used for validating. @param accountID the account id which needs to be validated against. @param response @param data the business context data to be used in validating. @return boolean @throws FatalException
IChannelDSL Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDSLAdapter	executeEmailTransaction	The following method executes the execute() of ETextCommand, which executes the email transaction. @param sessionContext @param businessContextDTO @param conversationText @return ContextResponse @throws FatalException	This method executes the email transaction. @param currentTransactionContext @param businessContextDTO @param transactionStatus @return ContextResponse @throws FatalException
IChannelDSL Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDSLAdapter	executeEmailTransaction	The following method executes the execute() of ETextCommand, which executes the email transaction. @param sessionContext @param businessContextDTO @param conversationText @return ContextResponse @throws FatalException	This method returns the instance of channel adapter. @return ChannelDSLAdapter class instance

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ChannelDSLAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\ChannelDSLAdapter	executeEmailTransaction	The following method executes the execute() of ETextCommand, which executes the email transaction. @param sessionContext @param businessContextDTO @param conversationText @return ContextResponse @throws FatalException	This method prepares the {@link BusinessContextDTO} to be used in calling the ChannelDSLAdapter. It then executes the email transaction through the adapter call and sets the response. @param sessionContext the session context to be used. @param channelContext the channel context to be used for transaction. @param transactionType @param emailTransactionContext contains all the information required to execute the email transaction. @return contextResponse contains information to be used to prepare the final response, which is sent to the user later. @throws BusinessException @throws FatalException
IPricingValidatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingValidatorAdapter	validateBusinessPolicy	This method will take pricingDomainObjectDTO as an input parameter. It will contain all the validations to be applied to the this particular Adapter class. It will check for all the validations that are kept and if the Validation fails it will add the validation error to a list of validation errors. once all the validations are checked it will then throw a validation exception passing the list of validation errors. This will be caught as a RuntimeException.	This method takes domain , domainCategory and pricingDomainObjectDTO as input parameters . It creates a unique instance of pricing adapter factory. It then calls the getPricingValidatorAdapter method of the pricing adapter factory passing domain , domain category and domain object as arguments. Pricing adapter factory will then return an instance of a particular pricing validator adapter based on the arguments passed . Then this method will call the validatebusinesspolicy method of that particular pricing validator adapter by passing the pricing domain object DTO as the argument. @param domain @param domainCategory @param pricingDomainObjectDTO
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	deletePartyAccountRelationship		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method will add PayeeName if it is null from portal side. @param sessionContext @param dto @param componentPreferenceDTO @param accountService @throws FatalException
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDT O		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDT O		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDT O		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDT O		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDT O		This method fetches the account Name @return String,accountName, It returns the account name.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipDTO		This method fetches the account Name @return String,accountName, It returns the account name.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchAllPartiesOfAccount		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchAllPartiesOfAccount		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchAllPartiesOfAccount		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchAllPartiesOfAccount		This method retrieves the owner partyId for the given account number. @throws FatalException @return,String, Returns the party Id.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPrimaryPartyOfAccount		This method retrieves the owner partyId for the given account number. @throws FatalException @return,String, Returns the party Id.
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	validatePartyId		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipAccessTemplate		
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipType		This methods validates the party account relationship.  It first fetches the userDetails of the logged in user and hence  extracts the partyId of the user. Using this partyId and the accountId  fetched from fetchTransactingAccountNumber, this method gets the required  party account relationship. 
IPartyAccountRelationshipApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IPartyAccountRelationshipApplicationServiceAdapter	fetchPartyAccountRelationshipType		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isExistingAccount		<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link DDStatementRequestDTO},ddStatementRequestDTO , It contains all the information required to place a</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key : contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : StatementPeriodType MI. Denotes the period type of the input.</li> <li>period : Period MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : Date MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : Date must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing : FeeDetailsDTO contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType : StatementRequestType, MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus : StatementRequestStatus must be null. It will be set to pending.</li> <li>statementNo : Must be set to null for statement order. It will be set at statement generation.</li> <li>customerInitiatedRequest : CustomerInitiatedRequest Must be null.</li> </ul> <p>@return Response. It contains the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link DDStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isExistingAccount		<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order term deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,{@link TDStatementRequestDTO},tdStatementRequestDTO , , It contains all the information required to place a demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;</p> <p>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.&lt;/li&gt; &lt;/ol&gt;</p> <p>&lt;/br&gt;For a System initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt;</p> <p>&lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDueDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt;  &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.&lt;/li&gt;  &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null. Will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;  &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null&lt;/li&gt; &lt;/ol&gt;  @return {@link TDStatementRequestInquiryResponse}. It contains the {@link TDStatmentRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;  &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isExistingAccount		<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link LoanStatementRequestDTO},loanStatementRequestDTO,, It contains all the information required to place a</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending</li> <li>statementNo : Must be set to null for statement order.It will be set at statement generation.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null</li> </ul> <p>@return {@link LoanStatementRequestInquiryResponse}. It contains</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the {@link LoanStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS</a></li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR</a></li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li><a href="#">AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR</a></li> </ol> <p>@throws FatalException</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountAndFetchProduct	@param accountId @return @throws FatalException	This method fetches the product code for the input account Id. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountAndFetchProduct	@param accountId @return @throws FatalException	<p>This method performs all the validations required for CRUD operations. In case of the mode being ADDITION or MODIFICATION, following methods are called</p> <ul style="list-style-type: none"> <li>checkMandatoryAttributes()</li> <li>validateAccount()</li> <li>fetchAndValidateProductClass()</li> <li>validateForMaximumNominationsLimit()</li> <li>validateAllParties()</li> </ul> <p>In case of mode being INQUIRE or DELETION, following methods are called</p> <ul style="list-style-type: none"> <li>checkMandatoryAttributes()</li> <li>validateAccount()</li> </ul> <p>@param sessionContext @param accountNominationRequestDTO @param mode @throws FatalException @throws BusinessException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method validates the transaction currency, if the transaction currency is not in account currency then it will throw an exception (FC_AC_MFD_4): Multi fund deposit transaction is possible in account currency only. @fcb.param MI, MultiFundDepositBusinessPolicyDTO <li> {@link com.ofss.fc.app.account.multifunddeposit.policy.dto.MultiFundDepositBusinessPolicyDTO}</li>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will fetch the account details. @fcb.param MI, multiFundDepositBusinessPolicyDTO<li> {@link com.ofss.fc.app.account.multifunddeposit.policy.dto.MultiFundDepositBusinessPolicyDTO}</li> @return response, AccountInquiryResponse <li> {@link com.ofss.fc.app.adapter.account.dto.AccountInquiryResponse}</li>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountConfigurationDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will return the facility linked to the CASA or OD account. @param sessionContext @param accountId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will return the facility linked to the LOAN account. @param sessionContext @param accountId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receipt Details are fetched.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from CASA module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will return the primary account holder of the account, @param sessionContext @param accountId @param accountType @return String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used to populate the alert DTO for addition of new operating instruction for an account @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used to populate the DTO for logging the Alert On Collection of Adhoc Fee @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used to populate the DTO for logging the alert ,while modifying the operating instructions for an account. @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method is used to populate the DTO values for creation of accountaddressLinkage,modification of address account linkage and the deletion of address linkage @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method should be used by all internal modules for inquiring the account. This will avoid Interaction begin and close related calls. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	The following method is used to fetch account related information. The following method is used to avoid Interaction begin and close related operations. @param accountId @param accountType @return @throws FCRException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will fetch all the details related to facility Id. This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	@param accountTransactionReturn @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	@param accountTransactionReturn @return @throws FatalException	This method is responsible for levying Periodic Fees on the account in case bulking flag is true.   Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.   The bulking indicator and the frequency is setup in pricing module.   Once the fee details are populated by the caller method this method internally gives a call   to levyGiven which in turn will debit the account and intimate pricing.   @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	@param accountTransactionReturn @return @throws FatalException	<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: <ul style="list-style-type: none"> <li>Get the alternate account for fees posting for the account on which transaction is being posted.</li> <li>Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.</li> <li>If alternate account is maintained, it will be returned else original account is returned. The fees will be posted to the account returned.</li> <li>Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()}</li> <li>The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them.</li> <li>Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off.</li> <li>The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object.</li> <li>Instantiate PricingAdapterFactory.</li> <li>Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter}</li> <li>Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being </li></ul></pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>deducted.&lt;/li&gt; &lt;/li&gt;Return boolean value indicating success or failure to the caller module.&lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;/ul&gt;@param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.&lt;/li&gt; &lt;/li&gt;@return boolean:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;true: If the fee is deducted and intimated successfully at the pricing end.&lt;/li&gt;</li> <li>&lt;li&gt;false: If there is any failure in method execution.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;</li> <li>&lt;li&gt;Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumber} // As there might be more than one legs to financial transaction.&lt;/li&gt; &lt;/pre&gt; @param feeDetails</li> </ul>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	<pre>@param accountTransactionReturn @return @throws FatalException</pre>	<pre>&lt;pre&gt;This method is used to call demandDepositAccountService.debitFunds method. &lt;/ol&gt;Following are the steps for this method. &lt;/li&gt;Populate accountTransaction from accountTransactionAdapter using populateAccountTransactionAdapter method.&lt;/li&gt; &lt;/li&gt;Call demandDepositAccountService.debitFunds&lt;/li&gt; &lt;/li&gt;Set accountTransaction.transactionAmount to accountTransactionAdapter.transactionAmount&lt;/li&gt; &lt;/li&gt; &lt;/ol&gt; &lt;/pre&gt; @param accountTransactionAdapter @return @throws FatalException</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	@param accountTransactionReturn @return @throws FatalException	<pre>This method is used to call demandDepositAccountService.debitFunds method. <ol><li>Following are the steps for this method.</li><li>Populate accountTransaction from accountTransactionAdapter using populateAccountTransactionAdapter method.</li><li>Call demandDepositAccountService.debitFunds</li><li>Set accountTransaction.transactionAmount to accountTransactionAdapter.transactionAmount</li></ol> @param accountTransactionAdapter @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitFunds	@param accountTransactionReturn @return @throws FatalException	This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @param context @param accountID @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAlternateAccountIDForFeesPosting	@param accountId @param accountType @return @throws FatalException	This provides the alternate account for the primary account for fees charging. @param accountId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAlternateAccountIDForFeesPosting	@param accountId @param accountType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAlternateAccountIdForFeesPosting	@param accountId @param accountType @return @throws FatalException	This service will be used to log periodic fee action for an account for a date. It checks whether an instance of the account for that date and action is already present or not. If it is present, a new instance will not be created. This is because an account needs to be picked up only once in the periodic fee shell for an action. <pre>The method works as follows: &lt;li&gt; Create an instance of the repository by calling the service {@link PeriodicFeeRepository#getInstance} &lt;/li&gt; &lt;li&gt; Load the periodicFeeLog with the input details by calling the repository service {@link PeriodicFeeRepository#load}&lt;/li&gt; &lt;li&gt; If the record does not exist, then call the repository service {@link PeriodicFeeRepository#insert} with the input details. &lt;/li&gt; </pre> @param accountId @param actionDate @param periodicFeeActionType @param earmarkId @param pricingTicketNumber @param transactionReferenceNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	reverseDebitFunds	@param debitResponse @param accountTransactionAdapter @return @throws FatalException	<pre>This method is internally called by {@link DDAFeesManagementService#reverseLeviedFees()} when fees levied earlier have to be reversed. This method does the following &lt;ul&gt; &lt;li&gt;Reverse debit the fees levied earlier using {@link DemandDepositAccountService#reverseDebitFunds()}. &lt;/li&gt; &lt;li&gt;Intimate pricing about the reversed fees so as to updated the counters at the pricing end.&lt;/li&gt; &lt;/ul&gt; The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt;Get the alternate account for fees posting for the account on which reversal is being done.&lt;/li&gt; &lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.&lt;/li&gt; &lt;li&gt;If alternate account is

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>maintained, it will be returned else original account is returned. The reversal will be done on the account that is returned.</p> <ul style="list-style-type: none"> <li>Call the method <a href="#">DemandDepositAccountService#reverseDebitFunds()</a> to reverse debit the fees applied for the transaction.</li> <li>Instantiate PricingAdapterFactory.</li> <li>Call <a href="#">PricingAdapterFactory#getPricingAdapter()</a> to get the pricing adapter instance of type <a href="#">IPricingAdapter</a>.</li> <li>Call <a href="#">PricingAdapter#intimateReversalToPricing(accountTransaction.transactionReferenceNumber: String; boolean)</a> to intimate the pricing module about the reversed fees. This will update the counters at the pricing end.</li> </ul> <p>The following are the parameters taken by the method along with their significance</p> <ul style="list-style-type: none"> <li><code>@param debitResponse[]</code> : The debit response details of the original transaction. See <a href="#">DebitResponse</a> for more details.</li> <li><code>@param accountTransaction</code> : The Account transaction details.</li> <li><code>@return boolean</code></li> <li><code>true</code>: If the fee is reversed and intimated successfully at the pricing end.</li> <li><code>false</code>: If there is any failure in method execution.</li> </ul> <pre>@param debitResponse @param accountTransaction</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	reverseDebitFunds	@param debitResponse @param accountTransactionAdapter @return @throws FatalException	<pre>This method is used to call demandDepositAccountService.reverseDebitFunds method. &lt;ol&gt;Following are the steps for this method. &lt;li&gt;Populate accountTransaction from accountTransactionAdapter using populateAccountTransactionAdapter method.&lt;/li&gt; &lt;li&gt;DeSerialized the debitResponse.&lt;/li&gt; &lt;li&gt;Call demandDepositAccountService.reverseDebitFunds&lt;/li&gt; &lt;li&gt;Set accountTransaction.transactionAmount to accountTransactionAdapter.transactionAmount&lt;/li&gt; &lt;/ol&gt; </pre> @param debitResponse @param accountTransactionAdapter @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to throw a overridable business policy error in case the party associated with the account is KYC incomplete or KYC unconfirmed,it first checks from the restricted services if there is any entry present for user defined status or system defined status of the account,it also checks if the account is debit blocked only and has service code and overridable flag set as true in FLX_AC_SERVICES_B to set validation errors. @param serviceId @param agingAccountStatus @param demandDepositLifecycleStatusList @param relationText TODO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will be used to calculate the interest benefit for the offset account case.  The interest benefit would be calculated for the capitalization cycle.  The balances for the account during the interest benefit period would be sent by the caller module.  The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the account details of the given CASA account id from DDA adapter. @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method validates the transaction currency, if the transaction currency is not in account currency then it will throw an exception (FC_AC_MFD_4): Multi fund deposit transaction is possible in account currency only. @fcb.param MI, MultiFundDepositBusinessPolicyDTO <li> {@link com.ofss.fc.app.account.multifunddeposit.policy.dto.MultiFundDepositBusinessPolicyDTO}</li>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will fetch the account details. @fcb.param MI, multiFundDepositBusinessPolicyDTO<li> {@link com.ofss.fc.app.account.multifunddeposit.policy.dto.MultiFundDepositBusinessPolicyDTO}</li> @return response, AccountInquiryResponse <li> {@link com.ofss.fc.app.adapter.account.dto.AccountInquiryResponse}</li>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method provides financial and non financial transaction details within specified period on demand deposit account.  On financial or non financial transaction event, application stores transaction details in terms of {@link StatementItem}. The list of the financial or non financial events on which details are logged is fixed.  Financial transaction generally updates {@link PrincipalBalances} of the account for example Cash withdrawal ,Cash deposit, Interest settlement etc.Non financial transaction does not update balances but they update non financial attributes of the account such {@link DemandDepositLifecycleStatus} , account address. Transaction inquiry request may come from branch



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>banking. In case of branch banking operation executive may want to see transaction performed on particular account within specified period. . {@link TransactionInquiryDTO} contains start and end date .&lt;br&gt; &lt;br&gt;This method lists all the transactions where {@link TransactionDates#postingDate} according to filters applied . Please go through {@link StatementInquiryFilterInputDTO} for more details. &lt;br&gt; &lt;br&gt; Financial transaction details contains following details. &lt;ol&gt; &lt;li&gt;Transaction Amounts: Amount in different currencies. &lt;li&gt;Transaction key/Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction. &lt;li&gt;Service Charge Details: Pricing reference number or pricing ticket from which details can be asked from pricing module. &lt;li&gt;Transaction source details- channel and application user. &lt;li&gt;Transaction dates: Different dates indicating realization of transaction. &lt;li&gt;Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic. Aggregate functions have been applied on financial transactions like Total amount credited, Total amount debited in given period. &lt;/ol&gt; Non financial transaction contain following details &lt;ol&gt; &lt;li&gt;Event date : Date on which event is performed. &lt;li&gt;Event description : description of the event like "Credit Interest rate is changed from 5% to 4.5%" &lt;/ol&gt; &lt;p&gt; &lt;br&gt; Following chart shows example of sample transaction inquiry response. Input details are Account : DDA_1,Period : '1-Jan-2010' to '15-Jan-2010' &lt;table border=1 &gt; &lt;tr&gt; &lt;td&gt;&lt;b&gt; Account No &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt; Posting Date&lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Transaction &lt;/b&gt;&lt;/td&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;&lt;b&gt;Credit/Debit&lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Transaction channel&lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Cheque Number &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Transaction Amount&lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Current Balance&lt;/b&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;1-Jan-2010&lt;/td&gt; &lt;td&gt;Cash Deposit&lt;/td&gt; &lt;td&gt;Cr&lt;/td&gt; &lt;td&gt;Branch&lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;td&gt;1000 INR&lt;/td&gt; &lt;td&gt;5000 INR&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jan-2010&lt;/td&gt; &lt;td&gt;Cash Withdrawal&lt;/td&gt; &lt;td&gt;Dr&lt;/td&gt; &lt;td&gt;Branch&lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;td&gt;2000 INR&lt;/td&gt; &lt;td&gt;3000 INR&lt;/td&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;15- Jan-2010&lt;/td&gt; &lt;td&gt;Cheque deposit&lt;/td&gt; &lt;td&gt;Cr&lt;/td&gt; &lt;td&gt;Branch&lt;/td&gt; &lt;td&gt;101010&lt;/td&gt; &lt;td&gt;5000 INR&lt;/td&gt; &lt;td&gt;8000 INR&lt;/td&gt; &lt;tr&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; Total credit : 6000 INR&lt;br&gt; Total Debit : 2000 INR &lt;br&gt; &lt;br&gt; Non financial event can be displayed in following manner for same inputs. &lt;table border=1&gt; &lt;tr&gt; &lt;td&gt;&lt;b&gt; Account No &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt; Event Date &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt; Event description &lt;/b&gt;&lt;/td&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jun-2010&lt;/td&gt; &lt;td&gt;Credit Interest rate is changed from 5% to 7%&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jun-2010&lt;/td&gt; &lt;td&gt;Account branch is changed from Vile Parle to Andheri.&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; &lt;br&gt; @fcb.param MI, SessionContext, sessionContext, , captures the basic information about the interaction being initiated with the system by the user @fcb.param MI,TransactionInquiryDTO,transactionInquiryDTO,, It contains the following attributes: &lt;ol&gt; &lt;li&gt;accountId: String MI it is the id of the account for which DD transactions are inquired for.&lt;/li&gt; &lt;li&gt; statementPeriodType: {@link StatementPeriodType} </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>MI denotes the period type for which the inquiry is done</p> <p>statementInquiryFilterInputDTO: {@link StatementInquiryFilterInputDTO}, it contains the following attributes:</p> <ul style="list-style-type: none"> <li>statementInquiryFilterTypes: {@link StatementInquiryFilterType}[] it has all the filters that need to be applied.</li> <li>Optional startAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the start amount range.</li> <li>endAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the end amount range.</li> <li>checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.</li> <li>checkNumberEnd: String MI if cheque range filter is chosen. Denotes the end range.</li> <li>transactionDescription: String MI if transaction description filter is chosen. Denotes the description for which the inquiry is to be made</li> <li>transactionMnemonics: String MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made</li> <li>lastNumberOfTransactions: int. Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs.</li> <li>lastNumberOfDays: int. This denotes the last number of days for which the transactions are to be fetched. If this is set to zero, then default values will be picked up from statement parameters.</li> <li>startDate: {@link Date} Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Also if either of Last_X_Transaction or Last_N_Days filter are used, it can be null. It is a mandatory input for all other cases. &lt;/li&gt; &lt;li&gt; endDate : {@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.&lt;/li&gt; &lt;li&gt;isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.&lt;/li&gt; &lt;/ol&gt; @return {@link DDTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link DDStatementItemDTO}.Each DDStatementItemDTO can be of following type.</p> <p>&lt;ol&gt; &lt;li&gt; {@link DDFinancialStatementItemDTO}: Demand deposit financial Item.- Contains financial details. &lt;li&gt; {@link DDNonFinancialStatementItemDTO}: Demand deposit non financial Item.- Contains non financial details. &lt;li&gt; {@link DDConsolidatedFinancialStatementItemDTO}: Consolidated {@link DDFinancialStatementItemDTO}. Please refer {@link ConsolidationType}. &lt;/ol&gt; All the different types of DDStatementItemDTO are combined and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.&lt;br&gt;&lt;br&gt; The following sections indicate the completion of the transactions with reasons: &lt;br&gt;&lt;b&gt;Transaction Success&lt;/b&gt;</p> <p>&lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS} &lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;</p> <p>&lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#FA</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>TAL_INT_ERROR} &lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_FILTER_CANNOT_BE_USED_ERROR} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_AMOUNT_GREATER_THAN_END_AMOUNT} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_CHEQUE_NUMBER_START_GREATER_THAN_CHEQUE_NUMBER_END} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_FILTER_MISMATCH_ERROR} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR} &lt;/li&gt;</li> <li>&lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR} &lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	<p>@param accountId @param accountType @return @throws FatalException</p>	<p>This method provides financial and non financial transaction details for input filter provided on term deposit account. On financial or non financial transaction event, application stores transaction details in terms of {@link StatementItem}.</p> <p>The list of the financial or non financial events on which details are logged is fixed. Financial transaction generally updates balances of the account for example TD payin, redemption, Interest settlement etc. Non financial transaction does not update balances but they update non financial attributes of the account account</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>address. Transaction inquiry request may come from branch banking or direct banking. &lt;br&gt; &lt;ol&gt; &lt;li&gt;In case of branch banking operation executive may want to see transaction performed on particular account within specified period.&lt;/li&gt; &lt;li&gt;In case of direct banking ,customer may want to see transactions performed on account. {@link TransactionInquiryDTO} contains start and end date .This method lists all the transactions where {@link TransactionAmount#transactionDate} is greater or equal than {@link TransactionInquiryDTO#startDate} and {@link TransactionAmount#transactionDate} is less than or equal to {@link TransactionInquiryDTO#endDate} . Please refer {@link StatementInquiryFilterInputDTO} for more details .&lt;/li&gt; &lt;/ol&gt; &lt;br&gt; Financial transaction details contains following details. &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Transaction Amounts: Amount in different currencies.&lt;/li&gt;</li> <li>&lt;li&gt;Transaction key / Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction.&lt;/li&gt;</li> <li>&lt;li&gt;Service Charge Details: Pricing reference number or pricing ticket from which details can be asked from pricing module.&lt;/li&gt;</li> <li>&lt;li&gt;Transaction source details- channel and application user.&lt;/li&gt;</li> <li>&lt;li&gt;Transaction dates: Different dates indicating realization of transaction.&lt;/li&gt;</li> <li>&lt;li&gt;Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic.&lt;br&gt; &lt;/ol&gt;</li> </ul> <p>&lt;br&gt; Non financial transaction contain following details &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Event date : Date on which event is performed.&lt;/li&gt;</li> <li>&lt;li&gt;Event description : description of the event like "Interest rate is changed from 5% to 4.5%"&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @param</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>transactionInquiryDTO MI, {@link TransactionInquiryDTO}, transactionInquiryDTO, It contains the following attributes: &lt;ol&gt; &lt;li&gt;accountId: String MI it is the id of the account for which TD transactions are inquired for.&lt;/li&gt; &lt;li&gt;statementPeriodType: {@link StatementPeriodType} MI denotes the period type for which the inquiry is done&lt;/li&gt; &lt;li&gt;statementInquiryFilterInputDTO: {@link StatementInquiryFilterInputDTO}, it contains the following attributes: &lt;ol&gt; &lt;li&gt;statementInquiryFilterTypes: {@link StatementInquiryFilterType}[] it has all the filters that need to be applied.Optional&lt;/li&gt; &lt;li&gt;startAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the start amount range. &lt;/li&gt; &lt;li&gt;endAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the end amount range.&lt;/li&gt; &lt;li&gt;checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.&lt;/li&gt; &lt;li&gt;checkNumberEnd: String MI if cheque range filter is chosen. Denotes the end range.&lt;/li&gt; &lt;li&gt;transactionDescription: String MI if transaction description filter is chosen. Denotes the description for which the inquiry is to be made&lt;/li&gt; &lt;li&gt;transactionMnemonics: String MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made&lt;/li&gt; &lt;li&gt;lastNumberOfTransactions: int. Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs. &lt;li&gt;lastNumberOfDays: int. This denotes the last number of days for which the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>transactions are to be fetched. If this is set to zero, then default values will be picked up from statement parameters. &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;startDate :{@link Date} Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods. Also if either of Last_X_Transaction or Last_N_Days filter are used, it can be null. It is a mandatory input for all other cases.&lt;/li&gt; &lt;li&gt;endDate : {@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.&lt;/li&gt; &lt;li&gt;isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.&lt;/li&gt; &lt;/ol&gt; @return {@link TDTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link TDStatementItemDTO}.Each TDStatementItemDTO can be of following type. &lt;ol&gt; &lt;li&gt;{@link TDFinancialStatementItemDTO}: Term deposit financial Item.- Contains financial details. &lt;li&gt;{@link TDNonFinancialStatementItemDTO}: Term deposit non financial Item.- Contains non financial details. &lt;li&gt;{@link TDConsolidatedFinancialStatementItemDTO}: Consolidated {@link TDFinancialStatementItemDTO}. Please refer {@link ConsolidationType}. &lt;/ol&gt; All the different types of TDStatementItemDTO are combined and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.&lt;/br&gt;&lt;/br&gt;The following sections indicate the completion of the transactions with reasons: &lt;/br&gt;&lt;b&gt;Transaction</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Success</p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</a></li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</a></li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li><a href="#">{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_FILTER_CANNOT_BE_USED_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_AMOUNT_GREATER_THAN_END_AMOUNT}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_CHEQUE_NUMBER_START_GREATER_THAN_CHEQUE_NUMBER_END}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_FILTER_MISMATCH_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}</a></li> </ol> <p>@throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountConfigurationDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This represents the updateAccountCreditMatrix method of the entity.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This represents the fetchAccountCreditMatrix method of the entity.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This represents the closeAccountCreditMatrix method of the entity.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This represents the reopenAccountCreditMatrix method of the entity.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	<p>@param accountId @param accountType @return @throws FatalException</p>	<p>Customers may want to see the transactions that have occurred in their demand deposit account. For this they will have to submit a statement request. Every statement request that a customer makes is tracked as {@link DDStatementRequest} for casa accounts. The customer may also want to reorder the same statement request as before. To enable this facility, this method will fetch all the statement request that have been generated in the given period by system. From this, customer can then reorder any one , multiple or all of the requests present for his particular account. The method will not fetch statement request which are just reorders. &lt;/br&gt; Consider the following example: &lt;p&gt; Suppose there is a statement request for the period 'x' to 'y' with statement id 'z'. Now if the customer reorders this request, there will be one more statement request generated with 'z' as the reorder id. Lets say the new statement request id is 'newZ'. This statement id will not be fetched by this service. The same is true even if the reorder is bulk. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId , ,This parameter denotes the casa accountId for which the statement requests are to be fetched. @fcb.param MI,Period, period, ,This denotes the date range for which the statement requests are to be fetched. &lt;/br&gt;It contains: &lt;ol&gt; &lt;li&gt;beginDate: MI;{@link Date}. It denotes the start date from which the statement requests are to be fetched.It must not be after the endDate.&lt;/li&gt; &lt;li&gt;endDate: MI;{@link Date}. It denotes the end date to which the statement requests are to be fetched. It must not be before the start date.&lt;/li&gt; &lt;/ol&gt; @return {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DDStatementRequestArrayInquiryResponse} This response contains the system as well as customer initiated requests {@link DDStatementRequest} for the given period that are system initiated, assembled into {@link DDStatementRequestDTO}. It also contains the Transaction Status. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;</p> <p><b>Transaction Success</b> &lt;/br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; <b>Transaction Failure</b> &lt;/br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt;</p> <p>&lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}&lt;/li&gt; &lt;/ol&gt; &lt;/ul&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	<p>@param accountId @param accountType @return @throws FatalException</p>	<p>Customers may want to see the transactions that have occurred in their term deposit account. For this they will have to submit a statement request. Every statement request that a customer makes is tracked as {@link TDStatementRequest} for term deposit accounts. The customer may also want to reorder the same statement request as before. To enable this facility, this method will fetch all the statement request that have been generated in the given period by system. From this, customer can then reorder any one, multiple or all of the requests present for his particular account. The method will not fetch statement request which are just reorders. &lt;/br&gt; Consider the following example: &lt;p&gt; Suppose there is a statement request for the period 'x' to 'y' with statement id 'z'. Now if the customer reorders this request, there will be one more statement request generated with 'z' as the reorder id. Lets say the new statement request id is 'newZ'. This statement id will not be fetched by this service. The same is true even if the reorder is bulk. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId, , This parameter denotes the casa accountId for which the statement requests are to be fetched. @fcb.param MI,Period ,period , , This denotes the date range for which the statement requests are to be fetched. &lt;/br&gt;It contains: &lt;ol&gt; &lt;li&gt;beginDate: MI;{@link Date}. It denotes the start date from which the statement requests are to be fetched. It must not be after the endDate.&lt;/li&gt; &lt;li&gt;endDate: MI;{@link Date}. It denotes the end date to which the statement requests are to be fetched. It must not be before the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>startDate.&lt;/li&gt; &lt;/ol&gt; @return {@link TDStatementRequestArrayInquiryResponse} This response contains the requested {@link TDStatementRequest} for the given period that are system initiated, assembled into {@link TDStatementRequestDTO}. It also contains the Transaction Status. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;</p> <p>&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt;</p> <p>&lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}&lt;/li&gt; &lt;/ol&gt; &lt;/ul&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<p>Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. &lt;/br&gt;Consider the following example. &lt;/br&gt;AccountId: TestAccount001 &lt;/br&gt;Opening date: 22nd-feb-2008 &lt;/br&gt;Statements generated: &lt;table border='1'&gt; &lt;tr&gt; &lt;th&gt;Statement no.&lt;/th&gt; &lt;th&gt;Statement period start&lt;/th&gt; &lt;th&gt;Statement period end&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;nth&lt;/td&gt; &lt;td&gt;1-april-2008&lt;/td&gt; &lt;td&gt;30-april-2008&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1&lt;/td&gt; &lt;td&gt;22-feb-2008&lt;/td&gt; &lt;td&gt;29-feb-2008&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</p> <p>Suppose the current process date is 24th may 2008.&lt;/br&gt;&lt;/br&gt;Then the current period: 1-may-2008 to 24th-may-2008 &lt;/br&gt;Prior statement period is: 1-april-2008 to 30-april-2008 &lt;/br&gt;&lt;/br&gt;In case there are no statements generated till 24th may 2008, then &lt;/br&gt;Current statement period: 22nd-feb-2008(opening date) to 24th may 2008 &lt;/br&gt;Prior statement period:22nd-feb-2008(opening date) to 24th may 2008&lt;p&gt;This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as follows: &lt;ol&gt; &lt;li&gt;start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>be after the current process date. &lt;li&gt;end date: Its the current process date. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId , , It is the account id of the account for which statement period is to be fetched. @fcb.param MI,{@link com.ofss.fc.enumeration.account.AccountType},accountType, , It denotes the type of account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit. @return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons &lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will return the facility linked to the CASA or OD account. @param sessionContext @param accountId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will return the facility linked to the LOAN account. @param sessionContext @param accountId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the total count of the arrangements of a broker for all the modules which include CASA,OD,LOANS,TD and ORIGINATION. if fetchFromAgentDB is true then it fetches the arrangementIds from the agentArrangementLinkage DB for a given BrokerId Otherwise it calls each module to get the corresponding arrangementIds on the basis of party id and then checks from agentArrangementLinkage DB to get the count of only those arrangementIds which belong to the required broker. Finally the count of the arrangementIDs of each module is returned in the CountDTO.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receipt Details are fetched.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to fetch the Details which the user has queried for a particular module. First the queries submitIds and details are fetched from CASA module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiry Response. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6) Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the addressLinkage for an account and add those values to the response. @param sessionContext @param accountId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will return the primary account holder of the account, @param sessionContext @param accountId @param accountType @return String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will fetch DemandDepositAccount through adapter call to CASA module and get account opening date of account @param parentAccountId @param accountingEventDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is responsible for logging balance change action @param accountTransaction @param accountId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This service will be used to levy periodic fees on demandDepositAccount. In case the bulkingRequired indicator is set to True in e-pricing, the service will consolidate the periodic fee amount due for the date and levy the same. It will also update the periodicFeeAmount at the account level which will subsequently be used for interest computation. <pre>The method works as follows: <ul><li><li>Open a loop for the input.periodicPriceFeatureDTO <li><li>If the value of the input parameter input.isBulkingRequired = True, add the value of the fee amount in a local variable.</li> <li><li>Prepare the object DDAFeeRequestInfo with the input details {In case of bulking, the FeeDetails will have only 1 row, else there will be a separate row for each fee code}</li> </ul> </li> <li><li>Call the service {@this.checkAndlevyFees} with the DDAFeeRequestInfo prepared above. </li> <li><li>Open a loop for the FeeAmount in the return parameter of the above service. <li><li>Add the actually debited fee amount in a local variable. </li> </li> <li><li>Call {@link AlternateAccountService#fetchAlternateAccountIdFor FeesPosting()} to get the alternate account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> id.&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; If the alternate account exists, then load the alternate account, else load the input.account by calling the service {@link DemandDepositAccountRepository#load}&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; If alternate account exists, then call the service {@link DemandDepositAccount#increasePeriodicFeeAmount} for the alternate account, else call the service for the main account with the total debited fee amount calculated above. &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Call the service {@link DemandDepositAccountRepository#update} to update the new periodicFeeAmount at the account level. &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Raise an accounting event by calling the service {@IDomainAccountingAdapter#raiseAccountingEvent }&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Log an action {@link BalanceChangeAction} with the debited account, i.e. the alternate account or the main account as the case may be. &lt;/pre&gt; @param listOfPriceFeatureDTOs @param accountTransaction @param depositsFeeRequestInfo @param pricingTicketNumber </pre>
IAccountDD AAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \account\IAccountDDA Adapter	fetchAccou nt	@param accountId @param accountType @return @throws FatalException	This method will fetch DemandDepositAccount through adapter call to CASA module and sets branch code and status of account in accountingEventDTO @param parentAccountId @param accountingEventDTO
IAccountDD AAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \account\IAccountDDA Adapter	fetchAccou nt	@param accountId @param accountType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	Method will be used to inquire account currency based on product class. @param accountId @param productClass @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method populates the activity log for the Multi fund deposit transaction. @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	@param accountId @param object @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to populate the alert DTO for addition of new operating instruction for an account @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to populate the DTO for logging the Alert On Collection of Adhoc Fee @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to populate the DTO for logging the alert on Periodic Statement Generation on Printing @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to populate the DTO for logging the alert ,while modifying the operating instructions for an account. @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to populate the DTO values for creation of accountaddressLinkage,modification of address account linkage and the deletion of address linkage @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the account details of the given CASA account id from DDA adapter. @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to inquire an account memo. The method accepts a DTO object MemoDTO, and in turn calls the assembler {@link MemoAssembler#toDomainObject} to convert it into the Memo object.   Susequently, a call to the domain service {@link MemoService#fetchAccountMemo} is made which inquires the details from the database. The method further calls the assembler {@link MemoAssembler#fromDomainObject} to convert it into the response object. @param memoDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	Based on Account Type method will check the status of Account. If Account status is closed throw Business Exception. @param accountId @param accountType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	<p>@param accountId @param accountType @return @throws FatalException</p>	<p>&lt;p&gt;This method would prepare the OperatingInstructionPartyDetailsDTO.It works as follows: ;For each&lt;br /&gt; OperatingInstructionPartyDetailsDTO, if the operating instruction channel is Branch: Check the Branch signatory&lt;br /&gt; flag set in party account relationship. Check if the KYC for the party is confirmed. Check if the party has valid &lt;br /&gt; user access role(template/account role) defined in OES for the given service. If all the above is true,the party&lt;br /&gt; would be added to list of parties who can operate on the account.If the operating&lt;br /&gt; instruction channel is Internet: Check the Internet signatory flag set in party account relationship Check if the &lt;br /&gt; KYC for the party is confirmed Check if the party has valid user access role(template/account role) defined in&lt;br /&gt; OES for the given service. If all the above is true,the party would be added to list of parties who can operate&lt;br /&gt; on the account.If the operating instruction channel is Mobile Banking: Check the Mobile&lt;br /&gt; Banking signatory flag set in party account relationship Check if the KYC for the party is confirmed Check if the &lt;br /&gt; party has valid user access role(template/account role) defined in OES for the given service. If all the above is &lt;br /&gt; true,the party would be added to list of parties who can operate on the account.; &lt;/p&gt; @param sessionContext @param returnDTO @param operatingInstructionPartyDetailsDTOs @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will set the account role name for the given account role,so if the party relation is SOW/JAF/JAO then the corresponding name of the enum is set in account role name else the name from account role of party is set in account role name. @param authorisedSignatoryPartyDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will set the account role name for the given account role,so if the party relation is SOW/JAF/JAO then the corresponding name of the enum is set in account role name else the name from account role of party is set in account role name. @param authorisedSignatoryPartyDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is responsible for fetching the account based on the AccountType. If account type is not present throw business exception. @param accountId @param accountType @return AccountDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method updates the internet banking flag of multiple accounts on the basis of partyId and accountId. @param sessionContext @param accountRelationshipDTOs @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param  MI,PartyToAccountRelationshipDTO[],partyToAccountRelationshipDTO,  ,partyToAccountRelationshipDTO @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	Fetches the Account DTO based on the Account type. @param accountType @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method populates AccountingEventDTO which is required to raise accounting event. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, TransactionLogDTO, transactionLog <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}</li> @fcb.param MI, List, accountingEventAmountTagDetailsDTO @fcb.param MI, String, eventId @fcb.param MI, MessageLogItemDTO, messageLogItem <li>{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}</li> @return PCAccountingEventDTO <li>{@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will be used to log next interest benefit action after checking eligibility i.e., in case when it is not backdated Offset or is new charging. The steps are: <ol> <li>Get the {@link AccountDTO} with the accountId and {@link AccountType}. <li>If {@link AccountDTO#isHasOffsetAccount()} AND {@link AccountStatus#Closed}, then form the {@link ActionBenefitActionLogData} and call {@link AccountActionLoggerService#logOffsetBenefitAction(String, String, Date, ActionBenefitActionLogData)}. @param accountId @param response
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the account details of the given CASA account id from DDA adapter. @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the account details of the given Loan Account Id. @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method populates the transaction audit log item. The Transaction fields mapped with transaction audit item as follows: TransactionAuditItem.TransactionAuditItemKey.referenceNumber = Transaction.AccountTransaction.transactionReferenceNumber TransactionAuditItem.TransactionAuditItemKey.sequenceNumber = Transaction.AccountTransaction.sequenceNumber TransactionAuditItem.accountNumber = Transaction.AccountTransaction.accountId TransactionAuditItem.transactionDates = Transaction.AccountTransaction.TransactionDates TransactionAuditItem.productCode = Account.product.getProduct.getCode TransactionAuditItem.depositNumber = Account.TermDeposit.depositNumber TransactionAuditItem.renewalNumber = Account.TermDeposit.renewalNumber TransactionAuditItem.originalDepositNumber = Account.TermDeposit.originalDepositNumber TransactionAuditItem.userReferenceNumber = Transaction.AccountTransaction.userReferenceNumber TransactionAuditItem.debitCardNumber = if transaction is being performed by debit card number. TransactionAuditItem.fromAccountBranch = branch code from session context TransactionAuditItem.originalUserReferenceNumber= TODO TransactionAuditItem.partyId =



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Account.TermDeposit.getCustomerId  TransactionAuditItem.userId = get the teller id from the session context  TransactionAuditItem.lastApproverId = last approver from the session context.  TransactionAuditItem.transactionAmounts = Transaction.AccountTransaction..transactionAmounts  TransactionAuditItem.transactionAttributes = Transaction.AccountTransaction.transactionAttributes  TransactionAuditItem.Channel=Transaction.Channel  TransactionAuditItem.instrumentDetails = Transaction.instrumentDetails  TransactionAuditItem.ContraAccount.accountNumber =  FundTransferTransaction.contraAccountTransaction.accountId  TransactionAuditItem.ContraAccount.depositNumber=TODO  TransactionAuditItem.ContraAccount.renewalNumber=TODO  TransactionAuditItem.ContraAccount.originalDepositNumber=TODO  TransactionAuditItem.ContraAccount.contraBranch = FundTransferTransaction.contraAccountTransaction.account.getBranchId  TransactionAuditItem.ContraAccount.contraAccountType =  FundTransferTransaction.contraAccountTransaction.account.getAccountType @param transaction @param accountType will be set only if account is CASA TD or loan. @throws FatalException @throws IOException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will fetch the CASA account{@link DemandDepositAccount} and then will get the party id and branch id and then log the DDAAction using action logger {@link ActionLogger}.The rate change of EOD is executed because when pricing calls CASA the use case called is either same day or for future date.And in both the cases it is required to keep it in EOD as there are no impacts analysed till to move them into BOD.If requirement come for executing it in BOD we can safely move this into EOD. @param accountId @param actionDueDate
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	@see IPricingAccountAdapter#getBasicDetailsForAccount(String, String)
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to fetch the account details for input account number and account type.   <i>Both accountId and accountType are mandatory inputs to this service.</i>  Any one of the following can be supplied in AccountType: <ul> <li>Loan</li> <li>Term_Deposit</li> <li>CASA</li> <li>Credit_Card</li> </ul>This service returns details of the account which includes title, nick name, status, branch, currency, offer, opening date, holder name, balance details, business unit and other account specific data like clear balance, unclear balance in case of CASA accounts, disbursement, schedule, installment, maturity details in case of LOAN accounts. (For detailed information please refer {@link com.ofss.fc.app.adapter.account.dto.AccountDTO})    Additionally, this service checks if there are any memos attached to this account. If found, it will returned in the {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				AccountInquiryResponse#getStatus()#getReplyText() as "The account memo is present on 'account number'. Description: 'memo description'".  If agent is linked to the account, this service also returns the agent details like agentId, agentType, agentName, commissionCategoryType, email, phone (For more, please check {@link com.ofss.fc.app.adapter.account.dto.AgentDetailsDTO }) @param sessionContext @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method should be used by all internal modules for inquiring the account. This will avoid Interaction begin and close related calls. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to fetch the account details for input account number.   <i>AccountId is a mandatory input to this service.</i>  It will first inquire the type of account and then invoke {@link #fetchAccountByAccountType(SessionContext, String, AccountType)} to fetch account specific details. @param sessionContext @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	The following method is used to fetch account related information.The following method is used to avoid Interaction begin and close related operations. @param accountId @param accountType @return @throws FCRException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	The method will validate to check if the transaction amount requires transacting party details are not. It would also return all the operating instructions that are applicable for the transaction amount specified. <p> Incase the amount is more than the threshold amount, the method will also set the reply code to {@link ResponseCodeType#OVR_EXCEP_PRES_01}.</p> @param sessionContext @param arrangementCode @param transactionAmount @param eventId @param accountType @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This will raise accounting in case of direct credit to CASA during multi fund deposit transaction. @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, FundsExchangeResponse, fundsExchangeResponse <li> {@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @fcb.param MI, String, accountId @fcb.param MI, FundDepositResponse, fundDepositResponse <li> {@link com.ofss.fc.app.dda.dto.transaction.FundDepositResponse}</li> @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	Preperation of BankingFactContextDTO for adding facts. @param adapterDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method fetches the existing nomination details for the respective accountId (CASA) or a accountId and depositNumber(TD). Following are the steps to be followed in this method:-  <ol> <li>Create an instance of AccountNominationsResponse </li> <li>Fetch the transaction status using fetchTransactionStatus().</li> <li>Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments</li> <li>Call the validate method</li> <li>If account type is Demand Deposit Account, then check for the existence of the account by calling fetchBasicDetails() of DemandDepositAccountInquiryService through the DDA adapter. Fetch the productId by calling fetchDemandDepositAccount through the DDAAdapter.(Done in validateAccountAndfetchProductCode )</li> <li>If account type is TermDeposit Account, then check for the existence of the account by calling fetchAccount() of TermDepositAccountApplicationService through the TD adapter. Validate deposit number and its status by calling fetchDeposit of DepositApplicationService through the TD adapter. (Done in validateAccountAndfetchProductCode )</li> <li>Create an object of Assembler</li> <li>Invoke its toDomainObject() method and convert the AccountNominationRequestDTO to the domain object</li> <li>Create an instance of Domain Service and invoke it's fetch method by passing NominationStructure. This would return a domain object.</li> <li>Invoke the fromDomainObject() method of Assembler and pass the domain object to it. It would return the response DTO in response.</li> <li>Set the DTO as a part of Inquiry Response</li> <li>For each party in the domain object fetched call

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>fetchPartyDetails of PartyApplicationService through the adapter. Set each response in the Inquiry Response. &lt;li&gt;The inquiry response will not have any deposit details. &lt;li&gt;Fill the Transaction Status by calling the method fillTransactionStatus and pass the transactionstatus as an argument to it.&lt;/li&gt; &lt;li&gt;Set the response status as transaction status&lt;/li&gt; &lt;li&gt;Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks.&lt;/li&gt; &lt;li&gt;Close the interaction in finally block.&lt;/li&gt; &lt;li&gt;return the Response of this operation.&lt;/li&gt; &lt;/ol&gt;</p> <p>/ @param sessionContext @param accountNominationRequestDTO @return AccountNominationsResponse @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<p>This method performs all the validations required for CRUD operations. In case of the mode being ADDITION or MODIFICATION, following methods are called &lt;li&gt;checkMandatoryAttributes()&lt;/li&gt; &lt;li&gt;validateAccount()&lt;/li&gt; &lt;li&gt;fetchAndValidateProductClass()&lt;/li&gt; &lt;li&gt;validateForMaximumNominationsLimit()&lt;/li&gt; &lt;li&gt;validateAllParties()&lt;/li&gt; In case of mode being INQUIRE or DELETION, following methods are called &lt;li&gt;checkMandatoryAttributes()&lt;/li&gt; &lt;li&gt;validateAccount()&lt;/li&gt; @param sessionContext @param accountNominationRequestDTO @param mode @throws FatalException @throws BusinessException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used for updating the workitem with the party Id. @param billPayDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for one time bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for set up pf periodic bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method is used to fetched allowed relief types for populating the matrix. @param sessionContext @param accountId @param hardshipReliefTypes @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<p>&lt;ol&gt;The method populates the DepositsFeeRequestInfo which is then passed to the FeesManagementService as input. The FeeManagementService requires instance of DepositsFeeRequestInfo &lt;/ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;set DepositsFeeRequestInfo.ModuleType = TaskModuleType.CASA&lt;/li&gt;</li> <li>&lt;li&gt;load account by {@link fetchDemandDepositAccount(accountId: String, moduleType: ModuleType): IAccount} &lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.partyId = account.getPartyId(&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.accountId = accountTransaction.getAccountId&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.eventId = accountTransaction.getEventId(&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.currencyCode[0] = accountTransaction.getCurrency(&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.PricingDetail = {@link getPricingDetail(FeeDetails: feeDetails): PricingDetails pricingDetails}&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.EvaluationDate = AccountTransaction.TransactionDates.PostingDate&lt;/li&gt;</li> <li>&gt; &lt;li&gt;set depositsFeeRequestInfo.userReferenceNumber = accountTransaction.UserReferenceNumber&lt;/li&gt;</li> <li>&lt;li&gt;set depositsFeeRequestInfo.TransactionAmount = accountTransaction.TransactionAmount.AmountInAccountCurrency&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; @param accountTransaction @param moduleType @param feeAmounts @param priceTicketNumber @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				DepositsFeeRequestInfo
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method calculates the applicable fee for setting the instruction. @param sessionContext @param feeDetailsDTO @param accountEventCodes @param accountServiceCode @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	This method will fetch all the details related to facility Id.This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<ol>The method populates the DepositsFeeRequestInfo which is then passed to the FeesManagementService as input. The FeeManagementService requires instance of DepositsFeeRequestInfo </ol> <ol>The method set the TaskModuleType as follows: <li>if the moduleType == DDA then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.CASA</li> <li>if the moduleType == TD then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.TERM_DEPOSIT</li> <li>load account by {@link fetchDemandDepositAccount(accountId: String, moduleType: ModuleType): IAccount} <li>set depositsFeeRequestInfo.partyId = account.getPartyId()</li> <li>set depositsFeeRequestInfo.accountId = accountTransaction.getAccountId</li> <li>set depositsFeeRequestInfo.eventId = accountTransaction.getEventId()</li> <li>set depositsFeeRequestInfo.currencyCode[0] = accountTransaction.getCurrency()</li> <li>set

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> depositsFeeRequestInfo.PricingDetail = {@link getPricingDetail(FeeDetails: feeDetails): PricingDetails pricingDetails}&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.EvaluationDate = AccountTransaction.TransactionDates.PostingDate&lt;/li &gt; &lt;li&gt;set depositsFeeRequestInfo.userReferenceNumber = accountTransaction.UserReferenceNumber&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.TransactionAmount = accountTransaction.TransactionAmount.AmountInAc countCurrency&lt;/li &lt;/ol&gt; @param accountTransaction @param moduleType @param feeAmounts @param priceTicketNumber @return DepositsFeeRequestInfo </pre>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccount	@param accountId @param accountType @return @throws FatalException	<pre> &lt;ol&gt; Based on module type the corresponding account gets loaded.If module type = DDA then &lt;li&gt;create instance account of {@link DemandDepositAccount}&lt;/li&gt; &lt;li&gt;set the {@link DemandDepositAccountKey} with AccountTransaction.accountId&lt;/li&gt; &lt;li&gt;load DemandDepositAccount by {@link DemandDepositAccountRepository#load(demandDep ositAccount: DemandDepositAccount)}&lt;/li&gt; &lt;li&gt;return account instance&lt;/li&gt; &lt;/ol&gt; @param accountId @param moduleType </pre>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyAccountTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyAccountTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyAccountTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object {@link TitleModificationResponseDTO}. Add the array of</li> </ul> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account ids by calling toArray() on the arraylist. <li>Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyAccountTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	This method will fetch all the details related to facility Id.This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyStatementAttributes	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	This method processes system initiated statement request.It does the following: <ol> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus. </li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details. <li>Call the {@link IAccountDDAAdapter#modifyStatementAttributes()} to update statement details at account level. </li>The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request. </ol> @param currentStatementRequest @param demandDepositAccountDTO @param statementparametersDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyStatementAttributes	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	This method processes system initiated statement request. It does the following: <ol style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details.</li> <li>Call the {@link IAccountTDAdapter#modifyStatementAttributes()} to update statement details at account level.</li> </ol> The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	modifyStatementAttributes	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	This method processes system initiated statement request. It does the following: <ol style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details.</li> <li>Call the {@link IAccountDDAAdapter#modifyStatementAttributes()} to update statement details at account level.</li> </ol> The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountPreferences	@param sessionContext @param accountId @return @throws FatalException	This method is used to populate the DTO for logging the alert on Periodic Statement Generation on Printing @param accountId @param object @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountPreferences	@param sessionContext @param accountId @return @throws FatalException	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	placeUserInitiatedHold	@param holdAmount @param accountTransactionAdapter @return	
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	placeUserInitiatedHold	@param holdAmount @param accountTransactionAdapter @return	<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: <ul style="list-style-type: none"> <li>Get the alternate account for fees posting for the account on which transaction is being posted.</li> <li>Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.</li> <li>If alternate account is maintained, it will be returned else original account is returned.</li> <li>Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()}.</li> <li>The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them.</li> <li>Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off.</li> <li>The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object.</li> <li>Instantiate PricingAdapterFactory.</li> <li>Call </li></ul></pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter}&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being deducted.&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Return boolean value indicating success or failure to the caller module.&amp;lt;/li&amp;gt; &amp;lt;/ol&amp;gt; The following are the parameters taken by the method along with their significance &amp;lt;ul&amp;gt; &amp;lt;/li&amp;gt; @param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; @return boolean: &amp;lt;ul&amp;gt; &amp;lt;/li&amp;gt; true: If the fee is deducted and intimated successfully at the pricing end.&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; false: If there is any failure in method execution.&amp;lt;/li&amp;gt; &amp;lt;/ul&amp;gt; &amp;lt;/li&amp;gt; &amp;lt;/ul&amp;gt; &amp;lt;/li&amp;gt; Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumb er} // As there might be more than one legs to financial transaction.&amp;lt;/li&amp;gt; &lt;/pre&gt; @param feeDetails </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	placeSystemInitiatedHold	This method will place the system initiated hold on the deposit account. @param holdAmount @param accountTransactionAdapter @return int	This method is responsible for levying Periodic Fees on the account in case bulking flag is true.   Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.   The bulking indicator and the frequency is setup in pricing module.   Once the fee details are populated by the caller method this method internally gives a call   to levyGiven which in turn will debit the account and intimate pricing.   @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @paramactionDate @paramperiodicFeeActionType @paramearmarkID @paramtransactionReferenceNumber	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: <ul style="list-style-type: none"> <li>Get the alternate account for fees posting for the account on which transaction is being posted.</li> <li>Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.</li> <li>If alternate account is maintained, it will be returned else original account is returned. The fees will be posted to the account returned.</li> <li>Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()}</li> <li>The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them.</li> <li>Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off.</li> <li>The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object.</li> <li>Instantiate PricingAdapterFactory.</li> <li>Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter}</li> <li>Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being </li></ul></pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				deducted.</li> </li>Return boolean value indicating success or failure to the caller module.</li> </ol> The following are the parameters taken by the method along with their significance <ul> </li>@param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.</li> </li>@return boolean: <ul> </li>true: If the fee is deducted and intimated successfully at the pricing end.</li> </li>false: If there is any failure in method execution.</li> </ul> </li> </ul> </li> </ul> </li> </ul> Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumber} // As there might be more than one legs to financial transaction.</li> </pre> @param feeDetails
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param pricingTicketNumber @param transactionReferenceNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This method is responsible for managing periodic fees . At the time of account opening in addition to the rate details , fee details are also returned by e-Pricing. If there is a periodic fee setup in the Price Benefit Chart then the chargeDate(Date on which fee is to be levied) and then the computationDate(Date on which fee needs to be computed on) is returned by Pricing A corresponding Periodic Fee Action is logged based on these dates returned by e-Pricing @param adapterDTO @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				periodicEventTrackerDTOArray
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This service will be used to log periodic fee action for an account for a date. It checks whether an instance of the account for that date and action is already present or not. If it is present, a new instance will not be created. This is because an account needs to be picked up only once in the periodic fee shell for an action. <pre>The method works as follows: &lt;/li&gt; Create an instance of the repository by calling the service {@link PeriodicFeeRepository#getInstance} &lt;/li&gt; &lt;/li&gt; Load the periodicFeeLog with the input details by calling the repository service {@link PeriodicFeeRepository#load}&lt;/li&gt; &lt;/li&gt; If the record does not exist, then call the repository service {@link PeriodicFeeRepository#insert} with the input details. &lt;/li&gt; </pre> @param periodicFeeLog
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This service will be used to log periodic fee action for an account for a date. It checks whether an instance of the account for that date and action is already present or not. If it is present, a new instance will not be created. This is because an account needs to be picked up only once in the periodic fee shell for an action. <pre>The method works as follows: &lt;/li&gt; Create an instance of the repository by calling the service {@link PeriodicFeeRepository#getInstance} &lt;/li&gt; &lt;/li&gt; Load the periodicFeeLog with the input details by calling the repository service {@link PeriodicFeeRepository#load}&lt;/li&gt; &lt;/li&gt; If the record does not exist, then call the repository service {@link PeriodicFeeRepository#insert} with the input details. &lt;/li&gt; </pre> @param accountId @param actionDate @param periodicFeeActionType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param earmarkId @param pricingTicketNumber @param transactionReferenceNumber
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This service will be used to update periodic fee action for an account for a date. It checks whether an instance of the account for that date and action is already present or not. If it is present, a new instance will not be created. This is because an account needs to be picked up only once in the periodic fee shell for an action. <pre>The method works as follows: &lt;li&gt; Create an instance of the repository by calling the service {@link PeriodicFeeRepository#getInstance} &lt;/li&gt; &lt;li&gt; Load the periodicFeeLog with the input details by calling the repository service {@link PeriodicFeeRepository#load}&lt;/li&gt; &lt;li&gt; If the record does not exist, then call the repository service {@link PeriodicFeeRepository#insert} with the input details. &lt;/li&gt; </pre> @param periodicFeeLog
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPeriodicFeeAction	This method will log the periodic fee action. @param accountId @param actionDate @param periodicFeeActionType @param earmarkID @param transactionReferenceNumber	This service will be used for the deletion of PeriodicFeeActionItem. The method in turn calls the repository service {@link PeriodicFeeRepository#delete} to delete the record with the input details. @param periodicFeeLog
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountForOperatingInstructions	This method is used to validate the Account before assigning an Operating Instruction to it. @param sessionContext @param accountId @return @throws FatalException	This method checks the validity of the specified account number.   It is invoked by every maintenance method of this service to check whether the specified account is valid or whether it exists or not.  @param sessionContext @param accountId @param accountType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateGroupForOperatingInstructions	This method is used to validate the Group before assigning an Operating Instruction to it. @param sessionContext @param accountId @return @throws FatalException	This method checks the validity of the specified account number.   It is invoked by every maintenance method of this service to check whether the specified account is valid or whether it exists or not.  @param sessionContext @param accountId @param accountType @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateGroupForOperatingInstructions	This method is used to validate the Group before assigning an Operating Instruction to it. @param sessionContext @param accountId @return @throws FatalException	This method is used to validate the Group before assigning an Operating Instruction to it. @param sessionContext @param groupId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchGroupIfAccountMemberOfGroup	This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @exception FatalException	<pre>This method is used to provide the operating instruction details based on the transaction amount.  When a financial transaction being performed on an account is to be processed, Operation executive checks whether any operating instructions are being defined for the respective account or not.  The transaction currency and the account currency may differ.  Considering such scenarios, the transaction amount is first converted into Account currency.  After conversion, the transaction amount is compared with the defined amount slabs in the financial instructions and the operating instruction with suitable amount slab is fetched as a response. This method will accept account Id and transaction amount as input arguments.  The converted transaction amount (in account currency) will be passed to domain service.  Here, it is assumed that the transaction amount is already being converted into account currency.  Following are the steps to be followed in this method:-  <ol><li>Create an instance of Inquiry Response</li><li>Create an object of Assembler</li></ol>Invoke its

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>toDomainObject() method and convert the FinancialInstructionRequestDTO to the domain object FinancialInstructionRequest.&lt;/li&gt;&lt;/li&gt;Create an instance of Domain Service and invoke it's fetch method by passing FinancialInstructionRequest. This would return domain object. This domain object will have the instruction details as a response.&lt;/li&gt;&lt;/li&gt;Invoke the fromDomainObject() method of Assembler and pass the domain object to it. It would return the DTO in response.&lt;/li&gt;&lt;/li&gt;Set the DTO as a part of InquiryResponse&lt;/li&gt;&lt;/ol&gt;&lt;/pre&gt; @param sessionContext @param financialInstructionRequestDTO</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchGroupIfAccountMemberOfGroup	This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance. If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @exception FatalException	<pre>This method is used to provide the non financial operating instruction details. When a non-financial transaction being performed on an account is to be processed, Operation executive checks whether any operating instructions are being defined for the respective account or not. This method will accept account Id, transaction Id and party Id as input arguments. The instruction description maintained against the Non-financial operating instruction type would be used in displaying in the text fields for non-financial operating instruction. Following are the steps to be followed in this method:- Create an instance of Inquiry Response Create an object of Assembler Invoke its toDomainObject() method and convert the NonFinancialInstructionRequestDTO to the domain object NonFinancialInstructionRequest. Create an instance of Domain Service and invoke it's fetch method by passing FinancialInstructionRequest. This would return domain object. This domain object will have the instruction details as a response. Invoke

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the fromDomainObject() method of Assembler and pass the domain object to it. It would return the DTO in response. Set the DTO as a part of InquiryResponse; &amp;Fill the Transaction Status by calling the method fillTransactionStatus and pass the transactionstatus as an argument to it. Set the response status as transaction status &lt;/pre&gt; @param sessionContext @param nonFinancialInstructionRequestDTO</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchGroupMemberOfGroup	<p>This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance. If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @exception FatalException</p>	<p>This method is used to provide the non financial operating instruction details. When an inquiry based transaction is being performed on an account, the operation executive checks whether any operating instructions are being defined for the respective account or not. This method will accept account Id, transaction Id and party Id as input arguments. The instruction description maintained against the Inquiry operating instruction type would be used in displaying in the text fields for inquiry operating instruction. Following are the steps to be followed in this method:- Create an instance of Inquiry Response. Create an object of Assembler. Invoke its toDomainObject() method and convert the InquiryInstructionRequestDTO to the domain object InquiryInstructionRequest. Create an instance of Domain Service and invoke its fetch method by passing FinancialInstructionRequest. This would return domain object. This domain object will have the instruction details as a response. Invoke the fromDomainObject() method of Assembler and pass the domain object to it. It would return the DTO in response. Set the DTO as a part of InquiryResponse @param sessionContext @param nonFinancialInstructionRequestDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchGroupOfAccountMemberOfGroup	This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance. If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @exception FatalException	This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance. If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchGroupOfAccountMemberOfGroup	This method determines whether an Account is a member of group for a purpose: Operating Instruction Maintenance. If so, it returns the Group id to which Account belongs as a "Group Member". @param sessionContext @param accountId @exception FatalException	This method is being defined for Internet Banking / Direct banking Application. It checks whether Operating Instructions are being maintained for the specified account, as per the Direct Banking requirement. It derives the complete Instruction Details and checks whether any instruction exists or not which has Mode of Operation as 'ANY'.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateMemoAccountDates	@param accountId @param demanddeposit @param firstMemoStartDate @param finalMemoEndDate	<ul>This method is called when the memo is created. This method updates the account.firstMemoStartDate and finalMemoEndDate in Account. The dates has to be updated in the Account as looking at these dates, it will be decided at the account level if the account active memo is present or not. The algorithm is as follows. <li>Call {@link MemoRepository#fetchMinMaxDates(memo.getKey.getAccountId(), memo.getStartDate(), memo.getEndDate())}</li> <li><li>This method will return list containing a record with two values i.e. minimumStartDate and maximumEndDate;</li><li>Get the first and second element from the list. The list has first element as firstMemoStartDate and the next as the finalMemoEndDate.</li> <li>update memo dates in account by calling {@code com.ofss.fc.app.adapter.impl.account.IAccountAdapter.updateMemoAccountDates(String accountId, AccountType accountType, Date firstMemoStartDate,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Date finalMemoEndDate)}</li> </ul> @param memo
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateMemoAccountDates	@param accountId @param demanddeposit @param firstMemoStartDate @param finalMemoEndDate	<ul style="list-style-type: none"> <li>This method is called when the memo is updated. This method updates the account.firstMemoStartDate and finalMemoEndDate in Account. The dates has to be updated in the Account as looking at these dates, it will be decided at the account level if the account active memo is present or not. The algorithm is as follows. &lt;li&gt;Call {@link MemoRepository#fetchMinMaxDates(memo.getKey().getAccountId(), memo.getKey().getMemoNumber(), memo.getStartDate(), memo.getEndDate())}&lt;/li&gt; &lt;li&gt;This method will return list containing a record with two values i.e. minimumStartDate and maximumEndDate;&lt;/li&gt; &lt;li&gt;Get the first and second element from the list. The list has first element as firstMemoStartDate and the next as the finalMemoEndDate.&lt;/li&gt; &lt;li&gt;update memo dates in account by calling {@com.ofss.fc.app.adapter.impl.account.IAccountAdapter.updateMemoAccountDates(String accountId, AccountType accountType, Date firstMemoStartDate, Date finalMemoEndDate)}&lt;/li&gt; &lt;/ul&gt; @param memo</li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountBalance	@param accountID @return	<p>This service will be used to levy periodic fees on demandDepositAccount. In case the bulkingRequired indicator is set to True in e-pricing, the service will consolidate the periodic fee amount due for the date and levy the same. It will also update the periodicFeeAmount at the account level which will subsequently be used for interest computation.</p> <pre>The method works as follows: <ul style="list-style-type: none"> <li>Open a loop for the input.periodicPriceFeatureDTO</li> <li>If the value of the input parameter input.isBulkingRequired = True, add the value of the fee amount in a local variable.</li> <li>Prepare the object DDAFeeRequestInfo with the input details {In case of bulking, the FeeDetails will have only 1 row, else there will be a separate row for each fee code}</li> <li>Call the service {@this.checkAndlevyFees} with the DDAFeeRequestInfo prepared above.</li> <li>Open a loop for the FeeAmount in the return parameter of the above service.</li> <li>Add the actually debited fee amount in a local variable.</li> <li>Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.</li> <li>If the alternate account exists, then load the alternate account, else load the input.account by calling the service {@link DemandDepositAccountRepository#load}</li> <li>If alternate account exists, then call the service {@link DemandDepositAccount#increasePeriodicFeeAmount} for the alternate account, else call the service for the main account with the total debited fee amount calculated above.</li> <li>Call the service</li> </ul></pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				{@link DemandDepositAccountRepository#update} to update the new periodicFeeAmount at the account level. </li> </li> Raise an accounting event by calling the service {@link IDomainAccountingAdapter#raiseAccountingEvent}</li> </li> Log an action {@link BalanceChangeAction} with the debited account, i.e. the alternate account or the main account as the case may be. </pre> @param listOfPriceFeatureDTOs @param accountTransaction @param depositsFeeRequestInfo @param pricingTicketNumber
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAdapter	fetchAccountBalance	@param accountId @return	@param accountId @param object @return
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAdapter	fetchAccountBalance	@param accountId @return	
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAdapter	fetchAccountBalance	@param accountId @return	
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAdapter	increasePeriodicFeeAmount	This method will increase the periodic fee amount. @param accountId @param feesMoney	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	increasePeriodicFeeAmount	This method will increase the periodic fee amount. @param accountId @param feesMoney	This method is responsible for levying Periodic Fees on the account in case bulking flag is true.   Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.   The bulking indicator and the frequency is setup in pricing module.   Once the fee details are populated by the caller method this method internally gives a call   to levyGiven which in turn will debit the account and intimate pricing.   @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	increasePeriodicFeeAmount	This method will increase the periodic fee amount. @param accountId @param feesMoney	This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	increasePeriodicFeeAmount	This method will increase the periodic fee amount. @param accountId @param feesMoney	<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt;Get the alternate account for fees posting for the account on which transaction is being posted. &lt;/li&gt; &lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id. &lt;/li&gt; &lt;li&gt;If alternate account is maintained, it will be returned else original account is returned. &lt;/li&gt; &lt;li&gt;Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()} &lt;/li&gt; &lt;li&gt;The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them. &lt;/li&gt; &lt;li&gt;Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off. &lt;/li&gt; &lt;li&gt;The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object &lt;/li&gt; &lt;li&gt;Instantiate PricingAdapterFactory. &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter} &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>deducted.&lt;/li&gt; &lt;/li&gt;Return boolean value indicating success or failure to the caller module.&lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt;@param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.&lt;/li&gt; &lt;li&gt;@return boolean: &lt;ul&gt; &lt;li&gt;true: If the fee is deducted and intimated successfully at the pricing end.&lt;/li&gt; &lt;li&gt;false: If there is any failure in method execution.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;li&gt;Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumber} // As there might be more than one legs to financial transaction.&lt;/li&gt; &lt;/pre&gt; @param feeDetails</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	decreasePeriodicFeeAmount	This method will decrease the periodic fee amount, @param accountId @param feesMoney	<pre>&lt;pre&gt;This method is internally called by {@link DDAFeesManagementService#reverseLeviedFees()} when fees levied earlier have to be reversed. This method does the following &lt;ul&gt; &lt;li&gt;Reverse debit the fees levied earlier using {@link DemandDepositAccountService#reverseDebitFunds()}.&lt;/li&gt; &lt;li&gt;Intimate pricing about the reversed fees so as to updated the counters at the pricing end.&lt;/li&gt; &lt;/ul&gt; The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt;Get the alternate account for fees posting for the account on which reversal is being done.&lt;/li&gt; &lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> id.&lt;/li&gt; &lt;/li&gt;If alternate account is maintained, it will be returned else original account is returned. The reversal will be done on the account that is returned.&lt;/li&gt; &lt;/li&gt;Call the method {@link DemandDepositAccountService#reverseDebitFunds()} to reverse debit the fees applied for the transaction.&lt;/li&gt; &lt;/li&gt;Instantiate PricingAdapterFactory.&lt;/li&gt; &lt;/li&gt;Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter}&lt;/li&gt; &lt;/li&gt;Call {@link PricingAdapter#intimateReversalToPricing(accountTr ansaction.transactionReferenceNumber: String): boolean} to intimate the pricing module about the reversed fees. This will update the counters at the pricing end.&lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;/ul&gt; &lt;/li&gt;@param debitResponse[] : The debit response details of the original transaction. See {@link DebitResponse} for more details&lt;/li&gt; &lt;/li&gt;@param accountTransaction : The Account transaction details.&lt;/li&gt; &lt;/li&gt;@return boolean &lt;/ul&gt; &lt;/li&gt;true: If the fee is reversed and intimated successfully at the pricing end.&lt;/li&gt; &lt;/li&gt;false: If there is any failure in method execution.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param debitResponse @param accountTransaction </pre>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkAndReleaseHold	This method will release the earmark for the given earmark id. @param accountId @param earmarkId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkAndReleaseHold	This method will release the earmark for the given earmark id. @param accountId @param earmarkId	This method will check and release all the holds that are being made on the account. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkAndReleaseHold	This method will release the earmark for the given earmark id. @param accountId @param earmarkId	This service will be used to levy periodic fees on demandDepositAccount. In case the bulkingRequired indicator is set to True in e-pricing, the service will consolidate the periodic fee amount due for the date and levy the same. It will also update the periodicFeeAmount at the account level which will subsequently be used for interest computation. <pre>The method works as follows: <ul><li></li></ul> Open a loop for the input.periodicPriceFeatureDTO <ul><li></li></ul> If the value of the input parameter input.isBulkingRequired = True, add the value of the fee amount in a local variable.</li> <li></li> Prepare the object DDAFeeRequestInfo with the input details {In case of bulking, the FeeDetails will have only 1 row, else there will be a separate row for each fee code}</li> </ul> </li> <li></li> Call the service {@this.checkAndlevyFees} with the DDAFeeRequestInfo prepared above. </li> <li></li> Open a loop for the FeeAmount in the return parameter of the above service. <li></li> Add the actually debited fee amount in a local variable. </li> </li> <li></li> Call {@link AlternateAccountService#fetchAlternateAccountIdFor FeesPosting()} to get the alternate account id.</li> <li></li> If the alternate account exists, then load the alternate account, else load the input.account by calling the service {@link DemandDepositAccountRepository#load}</li> <li></li> If alternate account exists, then call the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>service {@link DemandDepositAccount#increasePeriodicFeeAmount} for the alternate account, else call the service for the main account with the total debited fee amount calculated above. &lt;/li&gt; &lt;/li&gt; Call the service {@link DemandDepositAccountRepository#update} to update the new periodicFeeAmount at the account level. &lt;/li&gt; &lt;/li&gt; Raise an accounting event by calling the service {@link IDomainAccountingAdapter#raiseAccountingEvent} &lt;/li&gt; &lt;/li&gt; Log an action {@link BalanceChangeAction} with the debited account, i.e. the alternate account or the main account as the case may be. &lt;/pre&gt; @param listOfPriceFeatureDTOs @param accountTransaction @param depositsFeeRequestInfo @param pricingTicketNumber</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkAndReleaseHold	This method will release the earmark for the given earmark id. @param accountId @param earmarkId	This method is used to check if there are any earmarks applied when the fee was computed.   At the time of Fee computation only if the feeChargeType(PD Level)is EOP_EARMARK_ONLINE a user   initiated hold is created and the earmarkId is stored in the PeriodicFeeActionItem table.   Here a check is done if there are any such earmarkId;s present . If there are then   the hold is released by calling EarmarkService.liftHold  
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAllPeriodicFeeActionItem	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber	This method gives periodic fee ticket numbers for a particular process date. @param accountId @param originalTransactionReferneceNumber @param processDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return List<PeriodicFeeActionItemDTO>	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAllPeriodicFeeActionItem	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber @return List<PeriodicFeeActionItemDTO>	This method will check and release all the holds that are being made on the account. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAllPeriodicFeeActionItem	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber @return List<PeriodicFeeActionItemDTO>	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber @return List<PeriodicFeeActionItemDTO>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAllPeriodicFeeActionItem	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber @return List<PeriodicFeeActionItemDTO>	This service will be used for the deletion of PeriodicFeeActionItem. The method will then call the repository service {@link PeriodicFeeRepository#delete} to delete the record with the input details. @param periodicFeeLog

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deletePeriodicFeeActionItem	This method will remove the periodic fees action logged. @param accountId @param actionType @param actionDate @param pricingTicketNumber	This method will check and release all the holds that are being made on the account. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deletePeriodicFeeActionItem	This method will remove the periodic fees action logged. @param accountId @param actionType @param actionDate @param pricingTicketNumber	This method will remove the periodic fees action logged. @param accountId @param actionType @param actionDate @param pricingTicketNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	chargeFeeForStatementRequest	@param feeDetailsDTO @param accountId @param demandDepositAccountEventCodes @param demandDepositAccountServiceCode	System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link DDStatementRequestDTO},ddStatementRequestDTO , , It contains all the information required to place a demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt;  &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;  &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.&lt;/li&gt; &lt;/ol&gt;  &lt;/br&gt;For a System initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>StatementRequestType},MI, Must be set to SystemInitiatedRequest.&lt;/li&gt;  &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.It will be set at statement generation.&lt;/li&gt;. &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null&lt;/li&gt;  &lt;/ol&gt; @return {@link DDStatementRequestInquiryResponse}. It contains the {@link DDStatmentRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt; &lt;/ol&gt;</li> </ul> <p>The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId</p>	<p>The method will loop through all the {@link ActionRequest} and check for the following: &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If {@link DDActionType#BalanceChangeAction} is present, remove the following actions: &lt;ol&gt;</li> <li>&lt;li&gt;{@link DDActionType#InActivityStatusCheck}&lt;/li&gt;</li> <li>&lt;li&gt;{@link DDActionType#DormancyStatusCheck}&lt;/li&gt;</li> <li>&lt;li&gt;{@link DDActionType#UnclaimedStatusCheck}&lt;/li&gt; &lt;/ol&gt;</li> <li>&lt;li&gt;If {@link DDActionType#BackdatedTransaction} is present, remove the same set of actions as in the case of BalanceChangeAction.&lt;/li&gt;</li> </ul> <p>On the new set of action requests, loop through and derive the least of all the setIds of each actions which is greater than the current set. set this along with the new set of action requests in an instance of {@link ActionSetPreprocessorResponse} and return the same &lt;/p&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStatus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;The method will internally call the method <a href="#">DemandDepositAccount#validateStatus</a> with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId</p>	<p>This method will validate the account id on the basis of the account Id. @param accountId @param accountType</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStatus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;The method will internally call the method <a href="#">DemandDepositAccount#validateStatus</a> with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId</p>	<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; &lt;li&gt;Validate the party by calling <a href="#">PartyApplicationService#fetchPartyDetails</a> by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each <a href="#">@TitleModification#accountId</a> to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call <a href="#">PartyAccountRelationshipApplicationService#fetchTypesForAccounts</a>. This method will return an array of <a href="#">@AccountTypeDTO</a> indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of <a href="#">@AccountTypeDTO</a> and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call <a href="#">DemandDepositAccountApplicationService#modifyAccountTitle</a> TD : Call <a href="#">TermDepositAccountApplicationService#modifyAccountTitle</a> LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object <a href="#">@TitleModificationResponseDTO</a>. Add the array of account ids by calling toArray() on the arraylist. &lt;li&gt;Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param titleModificationResponseDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId</p>	<p>This is the public method exposed in the class. If the value of the attribute domainCategory = CASA, it internally calls the private method validateAccountStatus to perform the validations.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId</p>	<p>This method will be used to validate the status of the account during collection of adhoc fee. The transaction is not permitted for the accounts in the following statuses. &lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt; &lt;/ol&gt;The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event as Collect Adhoc Fee. In case the current status of the account is not allowed for the event, the error will be added to the list of validation errors. @param accountId</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses.</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt;</li> <li>&lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list ofValidation errors. @param accountId @param ServiceId @param eventId</p>	<p>This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses. &lt;ol&gt; &lt;li&gt;Dormant - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Unclaimed - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Inactive - SystemDefinedStatus&lt;/li&gt; &lt;li&gt;Debit Block - UserDefinedStatus&lt;/li&gt; &lt;/ol&gt;The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list ofValidation errors. @param accountId @param ServiceId @param eventId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getDDAAAdhocFeeServiceId	This method will return DDA Adhocfee Service Id @return serviceId	This service will be used to charge adhoc fees for an account for a particular Adhoc fee purpose.   It will perform the necessary business validations based on the type of account and call the services to levy the fees.   <pre>The method works as follows :           <ul> <li><li> Create an instance of the business policy class by calling the service {@link AdhocFeePurposeBusinessPolicyFactory#getInstance} to create an instance of the business policy. </li> <li> Call the method {@link AdhocFeePurposeBusinessPolicy#validate} to perform the business validations. </li> <li> If the value of input.domainCategory = Loans, then call the service {@link IAccountLoansAdpater#levyFees } to levy the fees on the account. TODO: Farook Shaikh to get back with the service details. Else </li> <li> Open a loop for the input.feeDetails </li> <li> Call the service {@link FeesManagementService#checkAndLevyFees} with input.pricingTicketNumber and loop record.feeDetails to levy the fees on the account. </li> <li> Call the service {@link IDomainAccountingAdapter#raiseAccountingEvent} to raise accounting event. </li> </ul> </pre> @param accountId @param domainCategory @param pricingTicketNumber @param feeDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getDDAAdhocFeeEventId	This method will return DDA Adhocfee Event Id @return eventId	This service will be used to charge adhoc fees for an account for a particular Adhoc fee purpose.   It will perform the necessary business validations based on the type of account and call the services to levy the fees.   <pre>The method works as follows :         <ul> <li>&lt;li> Create an instance of the business policy class by calling the service {@link AdhocFeePurposeBusinessPolicyFactory#getInstance} to create an instance of the business policy. &lt;/li> <li>&lt;li> Call the method {@link AdhocFeePurposeBusinessPolicy#validate} to perform the business validations. &lt;/li> <li>&lt;li> If the value of input.domainCategory = Loans, then call the service {@link IAccountLoansAdpater#levyFees } to levy the fees on the account. TODO: Farook Shaikh to get back with the service details. Else &lt;/li> <li>&lt;li> Open a loop for the input.feeDetails &lt;/li> <li>&lt;li> Call the service {@link FeesManagementService#checkAndLevyFees} with input.pricingTicketNumber and loop record.feeDetails to levy the fees on the account. &lt;/li> <li>&lt;li> Call the service {@link IDomainAccountingAdapter#raiseAccountingEvent} to raise accounting event. &lt;/li> </ul> </pre> @param accountId @param domainCategory @param pricingTicketNumber @param feeDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountForOverrides	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param accountId	<p>The method will maintain &amp;quot;Predefined Account Operating Instructions&amp;quot; for Financial,Non Financial Transactions and Inquiry Transactions. For Financial transaction this will provide the user only relevant signature images based on the amount of the transaction as part of signature verification process in case the channel used in transaction is Branch,in case the channel is Internet or Mobile Banking the parties needed to authorize the transactions based on the amount of the transactions is provided to SMS module. The predefined account operating instructions for financial transactions will be amount based and maintained at the account level. Amount Slabs will be defined and a corresponding 'Mode of Operation' will be assigned to the Amount Slab. The number of signatures required per slab will also be maintained. The user can choose specific Party IDs who can sign/authorize for the particular amount slab. The predefined account operating instructions for non-financial transactions will be maintained at the account level. The predefined account operating instructions for inquiry transactions will be maintained at the account level. In case the channel for operating instruction has not been set,then it is defaulted to Branch. The various modes of operation available are: Any 'n'. Here 'n' will be the number of customer signatures required. Singly. Jointly. Signatory Set. User can maintain any number of amount slabs and assign the mode of operation to the amount slabs respectively Each of the amount slabs requires the user to maintain Party IDs depending upon the mode of operation selected. Following are the steps to be</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>followed in this method:-&lt;br /&gt; &lt;br /&gt; Check for Mandatory attributes by passing Operating Instruction DTO as argument to validateBeforeProcess&lt;br /&gt; IF type == &amp;quot;GROUP&amp;quot; THEN &lt;br /&gt; Call validateAccount() and pass GroupId, Type and AccountType as input arguments. &lt;br /&gt; ELSE &lt;br /&gt; Call validateAccount() and pass AccountId, Type and AccountType as input arguments. &lt;br /&gt; &lt;br /&gt; Create an object of Assembler &lt;br /&gt; Pass the OperatingInstructionDTO as an argument to toDomainObject() of Assembler and convert it to domain object. &lt;br /&gt; This would return the OperatingInstruction&lt;br /&gt; &lt;br /&gt; Create an object of OperatingInstructionService domain service. &lt;br /&gt; Call the addInstructionForAccount method of Domain Service and pass the OperatingInstruction to it as an argument. &lt;br /&gt; Call the {<a href="#">@link AccountActivityEventLogger#logActivityForOnline</a>} to log alerts to be sent to the customer. &lt;/p&gt; @param sessionContext @param operatingInstructionDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountForOverrides	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param accountId	<pre> &lt;p&gt;&gt;This method is used to modify the operating instruction details.&lt;br /&gt; It doesn't modify the Mode of Operation. If the mode of operation need to be changed, then the operating instruction is deleted first and new instruction is created with new mode of operation.&lt;br /&gt;&lt;br /&gt; Steps are as follows:&lt;br /&gt; &lt;br /&gt; Check for Mandatory attributes by passing Operating Instruction DTO as argument to validateBeforeProcess&lt;br /&gt; IF type == &amp;quot;GROUP&amp;quot; THEN ;&lt;br /&gt; Call validateAccount() and pass GroupId, Type and AccountType as input arguments. &lt;br /&gt; ELSE &lt;br /&gt; Call validateAccount() and pass accountId, Type and AccountType as input arguments. &lt;br /&gt; &lt;br /&gt; Create an object of Assembler&lt;br /&gt; Pass the OperatingInstructionDTO as an argument to toDomainObject() of Assembler and convert it to domain object. This would return the entity OperatingInstruction.&lt;br /&gt; Create an object of OperatingInstructionService domain service.&lt;br /&gt; Call modifyInstructionFromAccount method of Domain Service and pass domain object OperatingInstruction to it as an argument.&lt;br /&gt; </pre> @paramsessionContext @paramoperatingInstructionDTO
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountForOverrides	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param accountId	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param operatingInstructionDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountForOverrides	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param accountId	This method checks the account for following over-ridable conditions:- - If the customer is blacklisted. - If the memo is attached to the account. - If the customer has a memo. @param accountId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchRunningBalanceFromBalanceHistory	This method will return the running balance on the given date. Running balance is calculated as :<p> clearBalance + unclearedFunds - overlineAmount - todAmount - utilizedOverdraftLimit - utilizedAUFLimit </p> If BalanceHistoryItem is not found, Running balance is returned as zero. @param accountId @param effectiveDate @return	This method will return the running balance on the given date. Running balance is calculated as :<p> clearBalance + unclearedFunds - todAmount - utilizedOverdraftLimit - utilizedAUFLimit </p> If BalanceHistoryItem is not found, Running balance is returned as zero. @param accountId @param date @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	inquireGroupDetailsForOperatingInstructionsAlert	This method is used to fetch the Party Id for a Group.This method is invoked while raising the Alert for Operating Instructions (adding as well as modification). In case of a Group, if Operating Instruction is being assigned or existing Operating Instruction is modified, the Alert is raised. Party Id for the respective group is fetched to set it in Alerts Details. @param groupId @return @throws FatalException	This method is used to populate the alert DTO for addition of new operating instruction for an account @param accountId @param object @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	inquireGroupDetailsForOperatingInstructionsAlert	This method is used to fetch the Party Id for a Group.This method is invoked while raising the Alert for Operating Instructions (adding as well as modification). In case of a Group, if Operating Instruction is being assigned or existing Operating Instruction is modified, the Alert is raised. Party Id for the respective group is fetched to set it in Alerts Details. @param groupId @return @throws FatalException	This method is used to populate the DTO for logging the alert ,while modifying the operating instructions for an account. @param accountId @param object @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	inquireGroupDetailsForOperatingInstructionsAlert	This method is used to fetch the Party Id for a Group.This method is invoked while raising the Alert for Operating Instructions (adding as well as modification). In case of a Group, if Operating Instruction is being assigned or existing Operating Instruction is modified, the Alert is raised. Party Id for the respective group is fetched to set it in Alerts Details. @param groupId @return @throws FatalException	This method is used to fetch the Party Id for a Group.This method is invoked while raising the Alert for Operating Instructions (adding as well as modification). In case of a Group, if Operating Instruction is being assigned or existing Operating Instruction is modified, the Alert is raised. Party Id for the respective group is fetched to set it in Alerts Details. @param groupId @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addOffsetAccount	This method is used to add a casa account as an offset account.	This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addOffsetAccount	This method is used to add a casa account as an offset account.	This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addOffsetAccount	This method is used to add a casa account as an offset account.	This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addOffsetAccount	This method is used to add a casa account as an offset account.	This represents the addOffsetAccountConfiguration method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountConfigurationDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addOffsetAccount	This method is used to add a casa account as an offset account.	This represents the addMultipleOffsetAccountConfigurations method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountConfigurationDTO[], @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the updateOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the deleteOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the updateOffsetAccountConfiguration method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,offsetAccountConfigurationDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This represents the reopenOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateOffsetAccount		This method is used to update offset Account when disbursement is reversed through payout @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteOffsetAccount		This represents the deleteOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteOffsetAccount		This represents the deleteOffsetAccountConfiguration method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,offsetAccountConfigurationDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	populateDefaultLinkageDates	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId	This represents the reopenOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	populateDefaultLinkageDates	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param accountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	closeOffsetAccount		This represents the closeOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	reopenOffsetAccount		This represents the reopenOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateDemandDepositAccountForLinkage		This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateDemandDepositAccountForLinkage		@param sessionContext @param offsetAccountDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateDemandDepositAccountForLinkage		This represents the updateOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateDemandDepositAccountForLinkage		This method is used to update offset Account when disbursement is reversed through payout @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		This will validate whether offset account type is casa or not, as Offset account can only be a valid CASA Account. @param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		This will validate offset account linkage. As accounts with same currency type can only be added.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		This method will validate that the beneficiary account selected should not already be linked as an offset with some other beneficiary.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateOffsetAccount		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAsOfCounter	The method will return the current asOfCounter of the given account. @param accountId @return	The method will loop through all the {@link ActionRequest} and check for the following: <ol style="list-style-type: none"><li>If {@link DDActionType#BalanceChangeAction} is present, remove the following actions: <ol style="list-style-type: none"><li>{@link DDActionType#InActivityStatusCheck}</li><li>{@link DDActionType#DormancyStatusCheck}</li><li>{@link DDActionType#UnclaimedStatusCheck}</li></ol></li><li>If {@link DDActionType#BackdatedTransaction} is present, remove the same set of actions as in the case of BalanceChangeAction.</li></ol> <p>On the new set of action requests, loop through and derive the least of all the setIds of each actions which is greater than the current set. set this along with the new set of action requests in an instance of {@link ActionSetPreprocessorResponse} and return the same</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitDDAAccount	This method will debit DDA account during processing of settlement payout interaction @param applicationContext @param accountId	This method will process the disbursement instructions for the settlement payout instruction. It will interact with CASA module for the processing of the disbursement instruction via AccountDDAAdapter. @param applicationContext @param instructionId @param actionDate @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountBalances	The method will fetch the balances for the given offset account @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createDebitInstruction		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createDebitInstruction		This methods setup process for settlement Instructions for a Application ,Facility or a Account level. Account level facility can be for loan account or OD. Facility level instruction again can be for an OD or loan Account. Application level can be done from origination.This method first validates the input coming from the client.All mandatory inputs need to send from the cliennt.It will then calls the method generateInstructionId to generates a unique instruction Id.This instruction id will be same for an item. setComponentPreferenceId method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. Business policy will be validated. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createDebitInstruction		This method will update the settlement payout instruction already maintained for an account or for a facility. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createContributionInstruction		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createContributionInstruction		This methods setup process for settlement Instructions for a Application ,Facility or a Account level. Account level facility can be for loan account or OD. Facility level instruction again can be for an OD or loan Account. Application level can be done from origination.This method first validates the input coming from the client.All mandatory inputs need to send from the cliennt.It will then calls the method generateInstructionId to generates a unique instruction Id.This instruction id will be same for an item. setComponentPreferenceId method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. Business policy will be validated. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createContributionInstruction		This method will update the settlement payout instruction already maintained for an account or for a facility. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateDebitInstruction		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteDebitInstruction		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteDebitInstruction		This method delete the settlement payout instruction maintained for a account, facility or application level . @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteContributionInstruction		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deleteContributionInstruction		This method delete the settlement payout instruction maintained for a account, facility or application level . @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsCrossOverFlag	The method will check if the account has crossed over from credit to debit or from debit to credit balance status. @param accountId @return	This method calls pricing and returns the pricing ticket number. @param currencyCodes @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsCrossOverFlag	The method will check if the account has crossed over from credit to debit or from debit to credit balance status. @param accountId @return	This method calls pricing and returns the pricing ticket number. @param currencyCodes @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	performBalanceChangeAction	The method will call {@link DemandDepositAccountService#intimateInterestForComputation}. @param accountId @param effectiveDate @param isCrossOver @param transactionType @param transactionAmount	This method calls pricing and returns the pricing ticket number. @param currencyCodes @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	performBalanceChangeAction	The method will call {@link DemandDepositAccountService#intimateInterestForComputation}. @param accountId @param effectiveDate @param isCrossOver @param transactionType @param transactionAmount	This method calls pricing and returns the pricing ticket number. @param currencyCodes @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will loop through all the {@link ActionRequest} and check for the following: <ol><li>If {@link DDActionType#BalanceChangeAction} is present, remove the following actions: <ol><li>{@link DDActionType#InActivityStatusCheck}</li><li>{@link DDActionType#DormancyStatusCheck}</li><li>{@link DDActionType#UnclaimedStatusCheck}</li> </ol><li>If {@link DDActionType#BackdatedTransaction} is present, remove the same set of actions as in the case of BalanceChangeAction.</li> <p> On the new set of action requests, loop through and derive the least of all the setIds of each actions which is greater than the current set. set this along with the new set of action requests in an instance of {@link ActionSetPreprocessorResponse} and return the same </p></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#Statement} action and log {@link DDActionType#ODStatement} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#BundleBenefitExpiry} action and log {@link DDActionType#ODBundleBenefitExpiry} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#CreditInterestSettlement} action and log {@link DDActionType#ODCreditInterestSettlement} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#CreditInterestCapitalization} action and log {@link DDActionType#ODCreditInterestCapitalization} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#DebitInterestCapitalization} action and log {@link DDActionType#ODDebitInterestCapitalization} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	hasAlternateForSettlement	The method will return true if the account has a alternate account setup for either credit or debit interest settlement. @param accountId @return	The method will remove the {@link DDActionType#DebitInterestSettlement} action and log {@link DDActionType#ODDebitInterestSettlement} if the account is an OD account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchActiveArrearSummaryByType		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	writeOffArr earPartially		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	writeOffArr earFully		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanc eDetails		This method is used to fetch the balance of a DemandDepositAccount.   The balance here is the totalAmount considering both the credit and debit buckets.   eg :   Clear Balance : 1000   AUF Limit : 100   AUF Utilized : 50   OD Limit : 100   OD Utilized : 50   The amount returned will be   Clear + (AUF Limit - AUF Utilized) + (OD Limit - OD Utilized)   100 + 50 + 50 = 200  
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanc eDetails		@param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanc eDetails		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanc eDetails		<p>This is the method that will be called from transactions widget of dashboard. It will be used when request for one time bill payment will be made. </p> @param sessionContext @param billPaymentRequestDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanceDetails		This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBalanceDetails		This method will fetch all the details related to facility Id.This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		This will compute the price at the time of recommendation .This get called from settlement payout instruction from draft mode where account is not opened yet. @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		<pre> / &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call @link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) Method to acquire a @link PricingTicketDTO). &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling @link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO) &lt;/li&gt; &lt;li&gt; Instantiate an object of @link PricingEventContextDTO) &lt;/li&gt; &lt;li&gt; Set the @link PricingTicketDTO) obtained earlier in this @link PricingEventContextDTO) &lt;/li&gt; &lt;li&gt; Instantiate @link EventPriceApplicationService} &lt;/li&gt; &lt;li&gt; Call the method @link EventPriceApplicationService#getPriceOnEvent() </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>which returns <a href="#">{@link PriceFeatureDTO}</a> &amp;lt;/li&gt;&amp;lt;/li&gt;&amp;lt;/ul&gt; Copy the values from <a href="#">{@link PriceFeatureDTO}</a> to an array of <a href="#">{@link PricingDetail}</a> as follows: &amp;lt;ul&gt;&amp;lt;li&gt; feeAmount : <a href="#">PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</a> prices.(MoneyBagDTO)netAmount &amp;lt;/li&gt;&amp;lt;li&gt; feeDescription : <a href="#">PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</a> prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &amp;lt;/li&gt;&amp;lt;/ul&gt; nsfTreatment : <a href="#">PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</a> prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &amp;lt;/li&gt;&amp;lt;/ul&gt; &amp;lt;/li&gt;&amp;lt;/ul&gt; Set the various attributes of <a href="#">{@link PricingDetail}</a> with the attribute values present in <a href="#">{@link PriceFeatureDTO}</a> by iterating through the details in it. &amp;lt;/li&gt;&amp;lt;/ol&gt; The following are the parameters taken by the method along with their significance &amp;lt;ul&gt;&amp;lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See <a href="#">{@link DepositsFeeRequestInfo}</a> for more details. &amp;lt;/li&gt;&amp;lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See <a href="#">{@link PricingDetail}</a> for more details. &amp;lt;/li&gt;&amp;lt;/ul&gt; &amp;lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		<pre> / &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} Method to acquire a {@link PricingTicketDTO}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {@link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate an object of {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Set the {@link PricingTicketDTO} obtained earlier in this {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate {@link EventPriceApplicationService} &lt;/li&gt; &lt;li&gt; Call the method {@link EventPriceApplicationService#getPriceOnEvent()} which returns {@link PriceFeatureDTO} &lt;/li&gt; &lt;li&gt; Copy the values from {@link PriceFeatureDTO} to an array of {@link PricingDetail} as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;li&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt; Set the various attributes of {@link </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingDetail} with the attribute values present in {<a href="#">@link PriceFeatureDTO</a>} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {<a href="#">@link DepositsFeeRequestInfo</a>} for more details. &lt;/li&gt; &lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See {<a href="#">@link PricingDetail</a>} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		<p>/ &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {<a href="#">@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)</a>} Method to acquire a {<a href="#">@link PricingTicketDTO</a>}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {<a href="#">@link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO</a>} &lt;/li&gt; &lt;li&gt; Instantiate an object of {<a href="#">@link PricingEventContextDTO</a>} &lt;/li&gt; &lt;li&gt; Set the {<a href="#">@link PricingTicketDTO</a>} obtained earlier in this {<a href="#">@link PricingEventContextDTO</a>} &lt;/li&gt; &lt;li&gt; Instantiate {<a href="#">@link EventPriceApplicationService</a>} &lt;/li&gt; &lt;li&gt; Call the method {<a href="#">@link EventPriceApplicationService#getPriceOnEvent()</a>} which returns {<a href="#">@link PriceFeatureDTO</a>} &lt;/li&gt; &lt;/ol&gt; Copy the values from {<a href="#">@link</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PriceFeatureDTO} to an array of {@link PricingDetail} as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;/ul&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; Set the various attributes of {@link PricingDetail} with the attribute values present in {@link PriceFeatureDTO} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; #param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {@link DepositsFeeRequestInfo} for more details. &lt;/li&gt; &lt;li&gt; #return PricingDetail[] : The fee details for the given event. See {@link PricingDetail} for more details. &lt;/li&gt; &lt;/ul&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		<p>Method used to fetch fees based on recommendation pricing ticket number. &lt;br&gt; When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number. Following steps are performed . &lt;ol&gt; &lt;li&gt; Create New fee ticker number. &lt;li&gt; Create DDA pricing event context. &lt;li&gt; Call Pricing service with old recommendation and new fee ticket number.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				</ol> @param feeRequestInfo @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		This will calculate the price at the time of recommendation. @param context @param accountId @param eventId @param amount @param accountType @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. @param sessionContext @param accountId @param eventId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		This methods fetches the fee at Reco mode in origination. @param sessionContext @param accountId @param eventId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEvent		This method validate the execution date. @param branchCode @param executionDate @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createHardShipRelief		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	createHardShipRelief		This represents the addHardshipReliefRequest method of the entity. 1)Following steps are carried out in the method: 1)System Constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplcation is called via AccountLoanApdater, LoanHardhsipRelief whose approval type are "Pending" are returned from the called method. 5)If DDAHardsipReliefRequestDetails are present, then DDAHardsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.app.account.dto.hardshipprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateHardShipReliefRequest		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateHardShipReliefRequest		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchHardShipReliefByApplicationId		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchHardShipReliefByApplicationId		<p>This represents the fetchHardshipReliefRequest method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.app.account.hardshiprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchHardShipReliefByApplicationId		This method will apply the authorisation decision for hardship relief request. If the decision is Approved, request and reliefs would be marked APPROVED and STP reliefs will be applied. If the decision is Rejected, request and reliefs will be marked REJECTED. Whereas for decision being Returned, the request and reliefs will be marked RETURNED. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,hardshipApplicationId, ,hardshipApplicationId @fcb.param MI,HardshipReliefRequestStatus,hardshipReliefRequestStatus, ,hardshipReliefRequestStatus @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	processFeeRefundTransaction	This method will process the reverse fees which will be refunded to the corresponding CASA account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param accountNo @param reversalDetailsDTO @throws FatalException	This method will process the reverse fees which will be refunded to the corresponding CASA/LOAN account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param sessionContext @param accountNo @param reversalDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	processFeeRefundTransaction	This method will process the reverse fees which will be refunded to the corresponding CASA account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param accountNo @param reversalDetailsDTO @throws FatalException	This method will process the reverse fees which will be refunded to the corresponding CASA account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param sessionContext @param accountNo @param reversalDetailsDTO @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateDDAHardshipReliefDetails		This represents the updateDDAHardshipReliefDetails method of the entity which will be used to update HardShipReliefType in case of CHANGE_INTEREST_RATE relief type. @param applicationContext @param ddaHardShipReliefDetailsDTOs

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateDDAHardshipReliefDetails		This represents the updateHardshipReliefRequest method of the entity. Following steps are carried out in the method: 1)System constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplciation is called via AccountLoanApdater. 5)If DDAHardhsipReliefRequestDetails are present, then DDAHardhsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEventStrictlyAtReco		This will calculate the price at the time of recommendation. @param context @param accountId @param eventId @param amount @param accountType @return @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getPriceOnEventStrictlyAtReco		This methods fetches the fee at Reco mode in origination. @param sessionContext @param accountId @param eventId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateHardShipReliefStatus		<p>This method will be called from updateDDAHardshipReliefDetails to update the status of hardship relief request. </p> Mandatory inputs for Hardship Relief are:   <ol> <li>requestId   <li>reasonForRejection  <li>reliefStatusType  </ol> @param applicationContext @param requestId @param reasonForRejection @param reliefStatusType
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateHardShipReliefStatus		@param sessionContext @param requestId @param hardshipAccountType @param reliefStatusType @param reasonForRejection @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	This method will intimate the collections module about the system and the user defined status of the account.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	This method would intimate collections in case a account belongs to DDA. @param accountDDAAdapter @param accountCreditMatrixDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	@param accountAddressLinkage @param accountDDAAdapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	This method will actually invoke the collection module with the new offer details @param demandDepositAccount @param accountBalanceRateDTO
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	@param demandDepositAccount @param accountBalanceRateDTO
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateCollectionForAccount	This method will be called from the ArrearService services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO @param accountId @throws FatalException	This method would be used to intimate rate change to the collection module @param arrangementNumber @param netRateDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	<p>TODO: Change the signature of debit() to include reason code, earmark parameter id. This method debits funds from the demand deposit account as specified by the amountToDebit parameter. It validates the account status and then tries to acquire the funds by calling the funds acquisition service, which operates on the funds acquisition sequence by calling a factory. This method works as follows: &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Validates whether the status of the account allows funds to be debited by calling the {@link DemandDepositAccount#validateStatus()} method with the appropriate mnemonic code.&lt;br&gt;This method will throw an exception in case the account status is not valid.&lt;/li&gt;</li> <li>&lt;li&gt;if(postingDate == processDate) &lt;li&gt;oldBalances = getBalances().getCurrentBalances() &lt;li&gt;Else if (postingDate &gt; processDate) &lt;li&gt;oldBalances = getBalances().getFutureBalances() if(amountToDebit.getCurrency != accountCurrency) then throw an exception stating that invalid currency for transaction &lt;li&gt;Tries to acquire funds from the account by calling the {@link FundsAcquisitionService#acquireFunds()}.&lt;/li&gt;</li> <li>&lt;li&gt;Based on the response {@link FundsAcquisitionResponse} from the FundsAcquisitionService, the following happens. &lt;ol&gt;</li> <li>The NSF treatment works as follows: &lt;li&gt;Has to be validated in case of both online and batch(i.e periodic) SC posting. &lt;li&gt;In case of Debit Interest, the treatment is defaulted to ForceDebit. &lt;li&gt;In case of Online transactions, NSF Treatment is not considered. &lt;/ol&gt;</li> </ul> <p>&lt;pre&gt; If amountAcquired &amp;lt; amountRequested Then Inspect the NSFTreatment. If NSFTreatment = Reject Then Throw FatalException. (TODO: Decide the error code) Else If NSFTreatment</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> = Waive Then //TODO: Need to reverse whatever FAS has done. Call reverse method Do nothing and return (TODO: Does the waive need to be logged?) Else If NSFTreatment = Hold Then //TODO: Need to reverse whatever FAS has done. Call reverse method this.getBalances().placeHold(amountRequested,proces sDate,&amp;lt;ExpiryDate&amp;gt;,&amp;lt;inputReasonCode&amp;gt;,&amp; lt;inputParameterId&amp;gt;,&amp;lt;generate the description&amp;gt;) (TODO: Decide the parameters to send) debitResponse.fundsHeld += amountRequested Else If NSFTreatment = PartDebitPartHold this.getBalances().placeHold((amountRequested - fundsAcquired),processDate,&amp;lt;ExpiryDate&amp;gt;,&amp;lt;in putReasonCode&amp;gt;,&amp;lt;inputParameterId&amp;gt;,&amp;lt;gen erate the description&amp;gt;) (TODO: Decide the parameters to send) debitResponse.fundsHeld += (amountRequested - fundsAcquired) Else If NSFTreatment = PartDebitPartWaive (TODO: Does the waive need to be logged?) Else If NSFTreatment = ForceDebit if(account is liability only) then throw fatal exception end if //This situation will never arise for an asset account which has either OD (it will always go overline) or TOD. TODO : In what case Force debit is applicable. End If else return; &lt;/pre&gt; &lt;/li&gt; &lt;li&gt;Call doSanctityCheck() &lt;li&gt;if(postingDate == processDate) &lt;li&gt;newBalances = getBalances().getCurrentBalances() &lt;li&gt;Else if (postingDate &gt; processDate) &lt;li&gt;newBalances = getBalances().getFutureBalances() &lt;li&gt;prepareTransactionResponse() &lt;li&gt;TODO: Raise arrears by analyzing the funds acquisition response&lt;/li&gt; &lt;li&gt;Return debitResponse&lt;/li&gt; &lt;/ol&gt; After calling doSanctityCheck counter inProgressCount is set to 0 so that if any other transaction which will come after that will get </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>inProgressCount as 0 because hibernate load this class from cache. There is possibility to get counter as 0.</p> <p>@param amountInAccountCurrency @param nsfTreatment @param transactionReferenceNumber @param debitType @param financialEventType @param transactionDates @param messageType @param eventId @param serviceId @param transactionAttributes @param subsequenceNumber @param initiationDetails @param systemEarmarkTypes @return</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	<p>This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId</p>	<p>This method credits funds to the demand deposit account as specified by the amountToCredit parameter. It validates the account status and then tries to credit the funds. This method works as follows:</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Validates whether the status of the account allows funds to be credited by calling the {@link DemandDepositAccount#validateStatus()} method with the appropriate mnemonic code.&lt;br&gt;This method will throw an exception in case the account status is not valid.&lt;/li&gt; &lt;li&gt;if(postingDate == processDate) &lt;li&gt;oldBalances = getBalances().getCurrentBalances() &lt;li&gt;Else if (postingDate &gt; processDate) &lt;li&gt;oldBalances = getBalances().getFutureBalances() &lt;br&gt;if(amountToCredit.getCurrency != accountCurrency) then throw an exception stating that invalid currency for transaction &lt;li&gt;Apportion the funds towards various heads such as holds placed by the system, or towards funds availed due to OD/AUF/Overline/TOD and eventually towards the cash balance. Call {@link FundsApportionmentService#apportionFunds}.&lt;/li&gt; &lt;li&gt;TODO: Appropriate arrears by analyzing the funds apportionment response&lt;/li&gt; &lt;li&gt;Form the</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>response and return it. TODO.&lt;/li&gt; &lt;li&gt;Call doSanctityCheck() &lt;li&gt;if(postingDate == processDate) &lt;li&gt;newBalances = getBalances().getCurrentBalances() &lt;li&gt;Else if (postingDate &gt; processDate) &lt;li&gt;newBalances = getBalances().getFutureBalances() &lt;li&gt;prepareTransactionResponse() &lt;/ol&gt; After calling doSanctityCheck counter inProgressCount is set to 0 so that if any other transaction which will come after that will get inProgressCount as 0 because hibernate load this class from cache. There is possibility to get counter as 0. @param amountToCredit @param referenceNumber @param dates @param financialEventType @param messageType @param eventId @param serviceId @param transactionAttributes @param subsequenceNumber @param nsfTreatment @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method completely reverses the effects of a debit which has taken place on the account at an earlier point in time.  An object {@link DebitResponse} containing information about the manner in which the original debit acquired funds is passed as input to this method and thus facilitates the reversal. This method works as follows. <ol> <li>Prepare fundsAcquisitionResponse {@link FundsAcquisitionResponse} from originalDebitResponse. <li>Call reverse acquired method of Funds Acquisition Service.<li>fundsReverseResponse {@link FundsReverseResponse} = {@link IFundsAcquisitionService#reverseAcquiredFunds(fundsAcquisitionResponse)} <li>Call Funds apportionment service to apportion funds, with the amount which was not reversed in the above step.<li>fundsAvailable = {@link FundsReverseResponse#getAmountReversalRequested()} - {@link FundsReverseResponse#getAmountReversal()}.<li>fundsApportionmentResponse {@link FundsApportionmentResponse} = {@link IFundsApportionmentService#apportionFunds(fundsAvailable)}. <li>Call this.doSanityCheck() for sanitation of the balance buckets. TODO : Revisit.<li> //Construct and log the financial statement item  {@link StatementItemService#addDDFinancialStatementItem()} //TODO:  </ol> @param originalReferenceNumber @param originalDebitResponse @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method completely reverses the effects of a credit which has taken place on the account at an earlier point in time. It uses the originalReferenceNumber to get the details of the original credit. TODO: Detailed steps to be written here. @param originalReferenceNumber @param originalCreditResponse @param financialEventType @param referenceNumber @param messageType @param eventId @param serviceId @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method used to modify {@code systemDefinedStatus} and {@code userDefinedStatus} for demand deposit account. {@link AccountStatusChangeBusinessPolicy} depends on what whether we want to change system or user defined status or both. <ol> <li>If you want to change only user defined status you can pass parameters as below <ol> <li>userDefinedStatusSet = new user defined status set. <li>statusChangeType = {@link AccountStatusChangeType#UserDefinedStatus} <li>systemDefinedStatus = null </ol> <li>If you want to change only system defined status you can pass parameters as below <ol> <li>statusChangeType = {@link AccountStatusChangeType#SystemDefinedStatus} <li>systemDefinedStatus = new system defined status <li>userDefinedStatusSet = null </ol> <li>If you want to change both the status then two status with {@link AccountStatusChangeType#Both} <li>Set {@code isSystemInitiated} flag properly. System defined status change is depends on this flag. please refer {@link AccountStatusChangeBusinessPolicy} for more details. </ol> This method will perform following activities. <ol> <li>Call business policy <ol> <li>Call {@link AccountStatusChangeBusinessPolicyFactory#createPol

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>icyInstance()) . &lt;li&gt;Call validatePolicy method on instance returned in step1. &lt;/ol&gt; &lt;li&gt;if {@code statusChangeType} is {@link AccountStatusChangeType#Both} or {@link AccountStatusChangeType#SystemDefinedStatus}, call modifySystemDefinedStatus method. &lt;li&gt;if {@code statusChangeType} is {@link AccountStatusChangeType#Both} or {@link AccountStatusChangeType#UserDefinedStatus}, call modifyUserDefinedStatus method. &lt;/ol&gt; @fcb.param SystemDefinedStatus,newSystemDefinedStatus, , This parameter captures the new SystemDefinedStatus of DDA account.&lt;br&gt; @fcb.param Set&lt;UserDefinedStatus&gt;,newUserDefinedStatusSet,Th is parameter captures the new newUserDefinedStatusSet of DDA account. &lt;br&gt; @fcb.param MI, boolean,isSystemInitiated,This parameter captures the information whether request is raised by system or user. &lt;br&gt; @fcb.param MI, AccountStatusChangeType,statusChangeType,This enumeration is represents whether caller want &lt;br&gt; to change {@link SystemDefinedStatus} or {@link UserDefinedStatus} or both. system or user. &lt;br&gt;</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This represents the updateAccountCreditMatrix method of the entity.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This represents the closeAccountCreditMatrix method of the entity.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This represents the reopenAccountCreditMatrix method of the entity.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method would intimate collections in case a account belongs to Loans. @param accountDDAAdapter @param accountCreditMatrixDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method would intimate collections in case a account belongs to DDA. @param accountDDAAdapter @param accountCreditMatrixDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method modifies the address linkages for all the parties associated with the account. Steps to be performed are 1) Set the task code of the operation for Interaction 2)Create Transaction Context by passing SessionContext and ADDITION (as MaintenanceType) as an arguments 3) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAINquiryApplicationService with accountId as the input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with accountId as the input. 4) Call validateAccount() method with accountId and status as the input. Only accounts with status " regular" are allowed to have linkages maintained. If the account status is closed or

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>dormant , " Closed/Dormant Account " error message would be displayed. 5) Check for the validity of the each party and its account relationship by calling fetchPartyAccountRelationship() method exposed by the PartyAccountRelationshipApplicationService. If the party is not valid, "Invalid Party" error message is displayed. If the party does not have any relationship with the account, "No Relationship with the Account" error message is displayed. 6) i) Call validateCombinedStatementSelection() method with AccountAddressLinkageDTO as the input. If the validation fails, error message would be displayed. ii) Call validateHoldOption() method with AccountAddressLinkageDTO as the input.If the validation fails, error message would be displayed. 7) Create an object of AccountAddressLinkageAssembler. Convert the AccountAddressLinkageDTO to domain object using the AccountAddressLinkageAssembler. 8) Create an instance of Domain Service (AccountAddressLinkageService). 9) Call the modifyAccountAddressLinkage() method of the AccountAddressLinkageService with the instance of AccountAddressLinkage created in the above step. 10) Fill the Transaction Status by calling the method fillTransactionStatus. 11) Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks. 12) Close the interaction in finally block. 13) Return the transactionStatus object as response of this operation. @param accountAddressLinkageDTO @param sessionContext</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	<p>This method accepts the domain object to perform the offer swap action. It calls for validating the business policies. If all criteria are met, it makes a call to the demand deposit repository and the offer swap audit repository to persist the entities to the database.</p> @param sessionContext @param offerSwapDetails
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method will actually invoke the collection module with the new offer details @param demandDepositAccount @param accountBalanceRateDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method will be called to create and carry forward the existing UDM to new offer while offer swap. @param demandDepositAccount @param offerSwapDetails @param parentAccountId @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	@param demandDepositAccount @param accountBalanceRateDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method is to used to construct the narration for logging the rate change in the non financial statement item.  Also a check is made to check whether the rate has actually changed or not. If the previous rate and the new rate is same then no narration would be generated.  Also the rate change would be logged only for credit rate and for debit case only OD would be considered.  In case of tiered rates the rate change would be displayed .Only a generic message would be logged stating your interest rate has changed please contact bank for further details. @param arrangementNumber @param netRateDTOsAsOnDate @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				netRateDTOsOnPreviousDate @param eventCode @param serviceId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	intimateCollection	This method returns true if the account is checking and has recovery flag set to true to call collections module. @param accountId	This method would be used to intimate rate change to the collection module @param arrangementNumber @param netRateDTO
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	The method will loop through all the {@link ActionRequest} and check for the following: <ol><li>If {@link DDActionType#BalanceChangeAction} is present, remove the following actions: <ol><li>{@link DDActionType#InActivityStatusCheck}</li><li>{@link DDActionType#DormancyStatusCheck}</li><li>{@link DDActionType#UnclaimedStatusCheck}</li> </ol><li>If {@link DDActionType#BackdatedTransaction} is present, remove the same set of actions as in the case of BalanceChangeAction.</li> <p> On the new set of action requests, loop through and derive the least of all the setIds of each actions which is greater than the current set. set this along with the new set of action requests in an instance of {@link ActionSetPreprocessorResponse} and return the same </p> </ol>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	This method will validate the account id on the basis of the account Id. @param accountId @param accountType



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	This method is called by the public method modifyAccountTitles(). This method works as follows:  <li>Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. <li>Create a hashmap called accountTitleMap. <li>Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. <li>Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. <li>Create an arraylist of strings to contain all the accounts for which modification was successful. <li>Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. <li>Construct the response object {@link TitleModificationResponseDTO}. Add the array of account ids by calling toArray() on the arraylist. <li>Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	This is the public method exposed in the class. If the value of the attribute domainCategory = CASA, it internally calls the private method validateAccountStatus to perform the validations.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	This method will be used to validate the status of the account during collection of adhoc fee. The transaction is not permitted for the accounts in the following statuses. <ol> <li>Dormant - SystemDefinedStatus</li> <li>Unclaimed - SystemDefinedStatus</li> <li>Inactive - SystemDefinedStatus</li> <li>Debit Block - UserDefinedStatus</li> </ol>The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event as Collect Adhoc Fee. In case the current status of the account is not allowed for the event, the error will be added to the list of validation errors. @param accountId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	validateAccountStaus	This method is used to validate the Account Status whether it is Unclaimed,WrittenOff or Closed. @param accountId	This method will be used to validate the status of the account. The transaction is not permitted for the accounts in the following statuses. <ol> <li>Dormant - SystemDefinedStatus</li> <li>Unclaimed - SystemDefinedStatus</li> <li>Inactive - SystemDefinedStatus</li> <li>Debit Block - UserDefinedStatus</li> </ol>The method will internally call the method {@link DemandDepositAccount#validateStatus} with the event eventId. In case the current status of the account is not allowed for the event, the error will be added to the list of Validation errors. @param accountId @param ServiceId @param eventId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitLMIAandLegalFee		This method is used to @param applicationContext @param instructionId @param actionDate
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitLMIAandLegalFee		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getStmtFrequencyForCASAAccount	The method will be used to inquire {@link StmtFrequency } of CASA account. @param accountId @return stmtFrequency	This method will place the next statement order. The method follows the below algorithm: <ol> <li>Check if the status of the account is either closed or Inactive. In each case ordering of next statement is not needed. <li>Calculate the beginning of the next statement period as end date of the last statement period plus one. <li>Calculate the end date of the next statement period using the statement frequency from Statement parameters. <li>Calculate the statement generation date with deferred statement generation date. Deferred generation date will only be taken into account if the frequency is monthly and above. Other wise statement generation date will be the same as the end date calculated. <li>Get the pricing ticket number from the pricing module. <li>Assemble a {@link DDStatementRequest} with all these details and call {@link StatementRequestService#orderStatement(DDStatementRequest, String, String)} using the same. </ol> @param demandDepositAccountDTO @param statementparametersDTO @param statementGenTill @param nextStatementPeriodBeginDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getStmtFrequencyForCASAAccount	The method will be used to inquire {@link StmtFrequency } of CASA account. @param accountId @return stmtFrequency	This method will place the next statement order. The method follows the below algorithm: <ol> <li>Check if the status of the account is either closed or Inactive. In each case ordering of next statement is not needed. <li>Calculate the beginning of the next statement period as end date of the last statement period plus one. <li>Calculate the end date of the next statement period using the statement frequency from Statement parameters. <li>Calculate the statement generation date with deferred statement generation date. Deferred generation date will only be taken into account if the frequency is monthly and above. Other wise statement generation date will be the same as the end date calculated. <li>Get the pricing ticket number from the pricing module. <li>Assemble a {@link DDStatementRequest} with all these details and call {@link StatementRequestService#orderStatement(DDStatementRequest, String, String)} using the same. </ol> @param demandDepositAccountDTO @param statementParametersDTO @param statementGenTill @param nextStatementPeriodBeginDate
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAllAccountsOfFacility		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchAccountBranch		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will loop through all the {@link ActionRequest} and check for the following: <ol style="list-style-type: none"> <li>If {@link DDActionType#BalanceChangeAction} is present, remove the following actions: <ol style="list-style-type: none"> <li>{@link DDActionType#InActivityStatusCheck}</li> <li>{@link DDActionType#DormancyStatusCheck}</li> <li>{@link DDActionType#UnclaimedStatusCheck}</li> </ol> </li> <li>If {@link DDActionType#BackdatedTransaction} is present, remove the same set of actions as in the case of BalanceChangeAction.</li> </ol> <p>On the new set of action requests, loop through and derive the least of all the setIds of each actions which is greater than the current set. set this along with the new set of action requests in an instance of {@link ActionSetPreprocessorResponse} and return the same</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#Statement} action and log {@link DDActionType#ODStatement} if the account is an OD account.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#BundleBenefitExpiry} action and log {@link DDActionType#ODBundleBenefitExpiry} if the account is an OD account.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#CreditInterestSettlement} action and log {@link DDActionType#ODCreditInterestSettlement} if the account is an OD account.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#CreditInterestCapitalization} action and log {@link DDActionType#ODCreditInterestCapitalization} if the account is an OD account.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#DebitInterestCapitalization} action and log {@link DDActionType#ODDebitInterestCapitalization} if the account is an OD account.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchIsProportionateOrOffset		The method will remove the {@link DDActionType#DebitInterestSettlement} action and log {@link DDActionType#ODDebitInterestSettlement} if the account is an OD account.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isAlternateAccount	This will check whether this account is alternate of some other account. @param accountId @return	This method is used to create new Account based action set for demand deposit account based on {@link EpochType}. This method will perform following activities <ol style="list-style-type: none"> <li>Create object reference <code>AccountBasedActionSet accountBasedActionSet</code> <li>If input.epochType = EndOfDay then   accountBasedActionSet = new DDEODActionSet(); <li>else input.epochType = BeginningOfDay then   accountBasedActionSet = new DDBODActionSet(); <li> <code>accountBasedActionSet.setAccountId(input.accountId)</code> <li>return accountBasedActionSet; <li> @param accountId @param epochType</li> </li></li></li></li></li></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAdapter	fetchIsShareFacility		This method is used to create new Account based action set for demand deposit account based on {@link EpochType}.This method will perform following activities <ol style="list-style-type: none"> <li>Create object reference <code>AccountBasedActionSet accountBasedActionSet</code> <code>&lt;/code&gt;</code> <li>If input.epochType = EndOfDay then   <code>accountBasedActionSet = new DDEODActionSet();</code> <li>else input.epochType = BeginningOfDay then   <code>accountBasedActionSet = new DDBODActionSet();</code> <li> <code>accountBasedActionSet.setAccountId(input.accountId)</code> <code>&lt;/code&gt;</code> <li>return accountBasedActionSet; <code>&lt;/ol&gt;</code>   @param accountId @param epochType</li></li></li></li></li></ol>
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAdapter	deposit		This method is used to modify the Primary reason and Secondary reason, when a request for Account Closure has been made. @param primaryReason @param secondaryreason
IAccountDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAdapter	deposit		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method processes system initiated statement request. It does the following: <ol style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>If it is found to be valid, then generate the statements using <a href="#">{@link #generateStatementRequest()} method</a>. Call the pricing module to charge for the service and update statement request generation details.</li> <li>Call the <a href="#">{@link IAccountDDAAdapter#modifyStatementAttributes()} method</a> to update statement details at account level.</li> </ol> The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		The following method is used to extract the depositNumber in case of a TD account
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will return all the details of a deposit that need to be displayed in the statement inquiry screen.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method takes care of all the consolidation details that is required and returns the inquiry response object filled with the statement items inquired. It follows the following steps: <ol style="list-style-type: none"> <li>Initialize sortedTDStatementItemDTOSet:TreeSet&lt;TDStatementItemDTO&gt; using the postingDateComparator</li> <li>Initialize a {@link StatementStatistics} object statementStatistics, and {@link TDStatementItemConsolidator} object tDStatementItemConsolidator</li> <li>Fetch the isChequeConsolidationRequired and isInteresAndTaxConsolidationRequired using the account product adaptor.</li> <li>Loop through all the financial statement items in {@link StatementInquiryResponse}. Do the following: <ol style="list-style-type: none"> <li>Fetch the consolidation type of the current item.</li> <li>If the type equals ConsolidationType.NONE, Assemble the item, add it to the sortedDDStatementItemDTOSet and update statementStatistics.</li> <li>If the type equals ConsolidationType.BULKFEE, add it to the dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.CHEQUE and isChequeConsolidationRequired, add it to dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.INTEREST_AND_TAX and isInteresAndTaxConsolidationRequired, add it to dDStatementItemConsolidator</li> </ol> </li> <li>Call the {@link TDStatementItemConsolidator#fetchStatementItems} and add the result list to sortedTDStatementItemDTOSet.</li> <li>Loop through all the all financial statement items in statementInquiryResponse and add it to sortedDDStatementItemDTOSet after assembling</li> </ol></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				them.</li> <li>Convert the TreeSet into {@link TDStatementItemDTO}[] and set it in the DDTransactionInquiryResponse along with statementStaticsics and return the same.</li> </ol> @param sessionContext @param tdTransactionInquiryResponseDTO @param statementInquiryResponse @param postingDateComparator @return TDTransactionInquiryResponse @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to populate set of {@link StatementItemFilter} from {@link TransactionInquiryDTO}. As {@link StatementInquiryService} works on set of filters.This method is used to fetch statement filters for demand deposit as well as term Deposit. @param accountDTO @see {@link StatementItemFilter} @param statementInquiryFilterInputDTOs @return @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Fetches the prior statement period @param sessionContext @param accountId @param depositId @param accountType @return @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Fetches the current Statement period. @param sessionContext @param accountId @param depositId @param accountType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method returns the transaction status for all the TD financial items. This method provides the feature of oracle text for fetching the information @param SessionContext sessionContext @param String accountId @param String queryPhrase @param PageInformationDTO pageInformation @return DDTransactionInquiryResponse @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		The following method is used to append the deposit id with the query phrase. @param depositId @param queryPhrase @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. </br>Consider the following example. </br>AccountId: TestAccount001 </br>Opening date: 22nd-feb-2008 </br>Statements generated: <table border='1'> <tr> <th>Statement no.</th> <th>Statement period start</th> <th>Statement period end</th> </tr> <tr> <td>nth</td> <td>1-april-2008</td> <td>30-april-2008</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1</td> <td>22-feb-2008</td> <td>29-feb-2008</td> </tr> </table> Suppose the current process date is 24th may 2008.</br></br>Then the current period: 1-may-2008 to 24th-may-2008 </br>Prior statement period is: 1-april-2008 to 30-april-2008 </br></br>In case there

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>are no statements generated till 24th may 2008, then  Current statement period: 22nd-feb-2008(opening date) to 24th may 2008  Prior statement period:22nd-feb-2008(opening date) to 24th may 2008</p> <p>This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as follows:</p> <ul style="list-style-type: none"> <li>start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not be after the current process date.</li> <li>end date: Its the current process date.</li> </ul> <p>@fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId , , It is the account id of the account for which statement period is to be fetched.</p> <p>@fcb.param MI,{@link com.ofss.fc.enumeration.account.AccountType},accountType, , It denotes the type of account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit.</p> <p>@return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li><b>Transaction Success</b></li> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> <li><b>Transaction Failure</b></li> <li>{@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated.  Consider the following example.  AccountId: TestAccount001  Opening date: 22nd-feb-2008  Statements generated: <table border='1'> <tr> <th>Statement no.</th> <th>Statement period start</th> <th>Statement period end</th> </tr> <tr> <td>nth</td> <td>1-april-2008</td> <td>30-april-2008</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1</td> <td>22-feb-2008</td> <td>29-feb-2008</td> </tr> </table> Suppose the current process date is 24th may 2008.  Then the current period: 1-may-2008 to 24th-may-2008  Prior statement period is: 1-april-2008 to 30-april-2008   In case there are no statements generated till 24th may 2008, then  Current statement period: 22nd-feb-2008(opening date) to 24th may 2008  Prior statement period:22nd-feb-2008(opening date) to 24th may 2008<p>This method will fetch the prior statement period for the given account id and account type. The prior statement period can be derived as follows: <ol> <li>start date: Its the start date of the latest system initiated statement request that as been

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>generated. If there are no such statement requests, then it will be the account opening date. &lt;li&gt;end date: Its the end date of the latest system initiated statement request that as been generated. If there are no such statement requests, then it will be the current process date. &lt;/ol&gt;</p> <p>@fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId , , It is the account id of the account for which statement period is to be fetched.</p> <p>@fcb.param MI,{@link com.ofss.fc.enumeration.account.AccountType},accountType , , It denotes the type of account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit.</p> <p>@return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons</p> <p>&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;</li></ul> <p>&lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt;</li> <li>{@link AccountErrorConstants#AC_NO_GENERATED_STATEMENT_FOUND}&lt;/li&gt;</li></ul> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to fetch the CASA account Details based on the accountIds. In this method the accountIds(arrangementIds) are passed which is used to fetch the CASA Account Details.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method handles {@link TDActionType#Statement} actions that have been logged. It is responsible for generating the statements that are part of the statement request period. The method follows the below steps: <ol style="list-style-type: none"> <li>Fetch the account details and the statement parameter details using the {@link IAccountDDAAdapter} and {@link IAccountProductAdapter} respectively.</li> <li>Get all the statement requests that need to be generated using {@link StatementRequestService#fetchTDStatementRequestOnActionDueDate(Date, String)} method.</li> <li>Loop through each of the statement request and do the following: <ol style="list-style-type: none"> <li>If the current statement request is reorder, then update the statement request with StatementGenerationStatus = ReOrdered and corresponding remarks. StatementStatisticsDTO must be null. Continue to the next request.</li> <li>If StatementRequestType = SystemInitiatedRequest then: <ol style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details.</li> <li>Call the {@link</li> </ol> </li> </ol> </li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IAccountTDAAdapter#modifyStatementAttributes()) to update statement details at account level. &lt;li&gt;The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request. &lt;/ol&gt; &lt;li&gt;For customer initiated statements,generate the statement using {@link #generateStatementRequest()} and update the statement request generation details. Continue to the next request. &lt;/ol&gt; &lt;li&gt;The same process will be repeated for all the remaining statement requests. &lt;/ol&gt; @param applicationContext @param actionKey @param actionRequest @param dueDate</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>Method will be used to remove statement action. @param nextDueDate @param accountId @param branchId</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method processes system initiated statement request.It does the following: <ol style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus. <li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details. <li>Call the {@link IAccountTDAAdapter#modifyStatementAttributes()} to update statement details at account level. <li>The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request. </li></li></li></li></ol> @param currentStatementRequest @param depositDetails @param statementparametersDTO @param adapterDepositDetailsDTOs2
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will process adhoc statement request.For customer initiated statements,generate the statement using {@link #generateStatementRequest()} and update the statement request generation details. Continue to the next request. @param currentStatementRequest @param isStatementWithoutAnyFinancialTransaction
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will process the statement requests that are reorders.If the statement request is reorder, then update the statement request with StatementGenerationStatus = ReOrdered and corresponding remarks. StatementStatisticsDTO must be null. @param currentStatementRequest @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				depositDetails
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method will place the next statement order. The method follows the below algorithm: &lt;ol&gt; &lt;li&gt;Check if the status of the account is either closed or Inactive. In each case ordering of next statement is not needed. &lt;li&gt;Calculate the beginning of the next statement period as end date of the last statement period plus one. &lt;li&gt;Calculate the end date of the next statement period using the statement frequency from Statement parameters. &lt;li&gt;Calculate the statement generation date with deferred statement generation date. Deferred generation date will only be taken into account if the frequency is monthly and above. Other wise statement generation date will be the same as the end date calculated. &lt;li&gt;Get the pricing ticket number from the pricing module. &lt;li&gt;Assemble a {@link TDStatementRequest} with all these details and call {@link StatementRequestService#orderStatement(IStatementRequest, String, String)} using the same. &lt;/ol&gt;</p> <p>@param productStatementParametersDTO @param stmtGenTill @param termDepositAccountDTO @param nextStatementPeriodBeginDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to validate the account status. If the System defined or User defined status of the current account has been marked as blocked at statement parameter level, then generation process will be halted. It returns the built up error message for each blocked status that is encountered. <p> For closed accounts, the generation process will be done only if the closure date falls in the period from the current statement period begin date till the end date plus deferred statement generation days if any. In this case it will be assumed to be the final statement request. </p> @param currentStatementRequest @param adapterDepositDetailsDTOs @param accountId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will update the {@link TDStatementRequest}. It sets the following: <ol> <li>StatementRequestStatus as Executed. <li>StatementStatistics in {@link StatementGenerationDetails}. <li>StatementGenerationStatus in {@link StatementGenerationDetails} if not null. This will be null in case the statement can not be generated. <li>Remarks in {@link StatementGenerationDetails}. </ol> @param statementRequest @param remarks @param inquiryResponse @param depositDetails
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will fetch the deposit details for the given account id and deposit number. @param accountId @param depositNumber @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method will set the statement opening and closing balance for the current statement period. &lt;p&gt;The opening balance for system initiated request will be the lastStatementClosingBalance stored at the account level. If this value is null, meaning its the first statement that is generated, then the opening balance will be stored as zero. &lt;/br&gt; In case of adhoc statements, opening balance will be the running balance of the last transaction that took place before the start date of the current statement request. &lt;/p&gt;&lt;p&gt;The closing balance for both system and adhoc request will be simply the running balance of the last transaction in the statement pried of the current request. If there are no transactions in the period, then the closing balance will be equal to the opening balance. &lt;/p&gt; @param statementRequest @param statementStatistics @param inquiryResponse @param depositDetails @return</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>@param priceTicketNumber @param feeAmount @param accountDebitResponse @param financialEventType @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This service will be used to levy periodic fees on demandDepositAccount. In case the bulkingRequired indicator is set to True in e-pricing, the service will consolidate the periodic fee amount due for the date and levy the same. It will also update the periodicFeeAmount at the account level which will subsequently be used for interest computation.</p> <pre>The method works as follows: <ul style="list-style-type: none"> <li>Open a loop for the input.periodicPriceFeatureDTO</li> <li>If the value of the input parameter input.isBulkingRequired = True, add the value of the fee amount in a local variable.</li> <li>Prepare the object DDAFeeRequestInfo with the input details {In case of bulking, the FeeDetails will have only 1 row, else there will be a separate row for each fee code}</li> <li>Call the service {@this.checkAndlevyFees} with the DDAFeeRequestInfo prepared above.</li> <li>Open a loop for the FeeAmount in the return parameter of the above service.</li> <li>Add the actually debited fee amount in a local variable.</li> <li>Call {@link AlternateAccountService#fetchAlternateAccountIdFor FeesPosting()} to get the alternate account id.</li> <li>If the alternate account exists, then load the alternate account, else load the input.account by calling the service {@link DemandDepositAccountRepository#load}</li> <li>If alternate account exists, then call the service {@link DemandDepositAccount#increasePeriodicFeeAmount} for the alternate account, else call the service for the main account with the total debited fee amount calculated above.</li> <li>Call the service</li> </ul> </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> {@link DemandDepositAccountRepository#update} to update the new periodicFeeAmount at the account level. &lt;li&gt; &lt;/li&gt; &lt;li&gt; Raise an accounting event by calling the service {@link IDomainAccountingAdapter#raiseAccountingEvent} &lt;/li&gt; &lt;li&gt; Log an action {@link BalanceChangeAction} with the debited account, i.e. the alternate account or the main account as the case may be. &lt;/pre&gt; @param listOfPriceFeatureDTOs @param accountTransaction @param depositsFeeRequestInfo @param pricingTicketNumber </pre>
IAccountDD AAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method is responsible for levying Periodic Fees on the account in case bulking flag is true. &lt;br&gt; Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated. &lt;br&gt; The bulking indicator and the frequency is setup in pricing module. &lt;br&gt; Once the fee details are populated by the caller method this method internally gives a call &lt;br&gt; to levyGiven which in turn will debit the account and intimate pricing. &lt;br&gt; @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException</p>
IAccountDD AAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will make feedetails out of depositFeeRequestInfo @param depositsFeeRequestInfo @return feeDetails
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: <ul style="list-style-type: none"> <li>&lt;li&gt;Get the alternate account for fees posting for the account on which transaction is being posted.&lt;/li&gt;</li> <li>&lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdFor FeesPosting()} to get the alternate account id.&lt;/li&gt;</li> <li>&lt;li&gt;If alternate account is maintained, it will be returned else original account is returned. The fees will be posted to the account returned.&lt;/li&gt;</li> <li>&lt;li&gt;Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()}&lt;/li&gt;</li> <li>&lt;li&gt;The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them.&lt;/li&gt;</li> <li>&lt;li&gt;Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or </li></ul></pre>





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method processes system initiated statement request. It does the following: &lt;ol&gt; &lt;li&gt;Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus. &lt;/li&gt;If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details. &lt;/li&gt;Call the {@link IAccountDDAAdapter#modifyStatementAttributes()} to update statement details at account level. &lt;/li&gt;The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request. &lt;/ol&gt;</p> <p>@param currentStatementRequest @param demandDepositAccountDTO @param statementPolicyDTO</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method used to create term deposit statement request. Customer or system can initiate statement request. statement request can be for demand deposit or term deposit account. System can initiate request using statement setup done at product/offering level or overridden a account level. Customer can ask for new statement or reorder existing statement . Statement request generally contain</p> <ul style="list-style-type: none"> <li>&lt;ol&gt;</li> <li>&lt;li&gt;Period - Start and End date</li> <li>&lt;li&gt;Delivery mode details</li> <li>&lt;li&gt;Service charge details.</li> </ul> <p>&lt;/ol&gt;</p> <p>&lt;p&gt;The method follows the following algorithm:</p> <ul style="list-style-type: none"> <li>&lt;ol&gt;</li> <li>&lt;li&gt;Set the statement request key from {@link UUIDGenerator#generateUuid()} &lt;li&gt;Set the statementRequestStatus as pending.</li> <li>&lt;li&gt;Create a new instance of {@link AbstractBusinessPolicy} based on the instance type of statementRequest and validate the same.</li> <li>&lt;li&gt;The statement requests will be placed at the deposit level as it an account level statement request may cause inconsistencies during batch. Hence for the current account, all the deposits are fetched and statement request is placed for each with {@link TDStatementRequest#accountBasedStmtId} as the same.</li> <li>&lt;li&gt;For System initiated request, make sure that the {@link StatementRequest#getCustomerInitiatedRequest()} is initialized with all the components as null so that isReorder flag is not persisted as null.</li> <li>&lt;li&gt;Call the {@link #create(AbstractDomainObject, com.ofss.fc.framework.repository.AbstractRepository)} method with instance of {@link StatementRequestRepository}</li> </ul> <p>&lt;/ol&gt;</p> <p>@param tdStatementRequest @param partyId, Denotes the partyId @param branchId, Denotes the branchId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will loop through all the deposit of the current account and log statement request for each. @param tdStatementRequest
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. This method will fetch the system initiated statement request that as been generated,for the given account which has the latest end date. In other words that latest generated statement request that was system initiated. If no such record is found, it returns null. It simply calls the {@link StatementRequestRepository#fetchLatestStatementPeriod} method and returns the same. Prior and current statement period calculations can be done based on this value. @param accountId @param accountType @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		As the statement requests are at account level , a new deposit will also need to be be part of statement generation. So we add a separate statement request for all new deposits on its opening. This method will loop through system initiated pending statement and add a separate request for the new deposit that has been opened. <p> It will return the statement request id of the newly added deposit request.If statement flag is not enabled, it will return null. </p> @param accountId @param depositNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>On deposit opening reversal, we need to remove any statement request that was logged. This method will remove all the statement requests for the given account id and deposit. &lt;p&gt; Note: Action removal will not be done. It is assumed that the caller will remove all the action. &lt;/p&gt; @param accountId,MI denotes the account Id of the account that is beaning closed @param depositNumber,MI deposit number which is to be closed. @return affectedRows ,The number of statement requests that are deleted.</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>On account closure, the customer may choose to get the statements till date for the current statement period to be generated and given to him instead of the usual statement generation date that is based on the statement frequency of his/her account. This method will log an statement request for this situation and also remove the already existing statement request that was logged as part of previous statement period generation. This method is for term deposit account statements. &lt;p&gt;The method does the following: &lt;ol&gt;  &lt;li&gt;Deletes the latest un-generated system initiated request by calling the {@link #deleteLastUnExecutedSystemDDStatementRequest()} method. &lt;li&gt;Creates a new statement request for the period from the last statement generated till date till the current process date. &lt;li&gt;If the last statement generated till date is null, then the statement request will be logged for the period between the account opening date till the current process date. &lt;/ol&gt;  &lt;/p&gt; @param accountId,MI denotes the account Id of the account that is beaning closed @param depositNumber,MI deposit number which is to be closed. @param productCode,MI denotes the product code of the account @param partyId,MI denotes the party id of the account @param</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>lastStatementGeneratedTill,Denotes the last statement generated till date.If this is null, the accountOpeningDate will be used. @param depositOpeningDate,Denotes the account opening date. It is mandatory if last statement generated till date is null. @param branchCode,Denotes the branch code of the deposit. @return returns true if the statement will be generated on the same day as the deposit closure.</p>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>TD redemption can be reversed. The statement request added or removed during closure have to be undone. This method will remove the statement request that will have been logged if the account closure statement is on the same day and add request for the current cycle. If account closure statement is not on the same day, then nothing needs to be done. @param accountId,MI denotes the account Id of the account that is bearing closed @param depositNumber,MI deposit number which is to be closed. @param productCode,MI denotes the product code of the account @param branchCode,MI denotes the branch code of the deposit. @param partyId,MI denotes the party id of the account @param depositOpeningDate,MI denotes the deposit opening date @param lastStatementGenTill,denotes the last statement generated till date. @return returns true if statement was added for the current statement cycle.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to delete a deposit statement request. Statement requests may need to be deleted if for system initiated cases, there is a change in the statement frequency or account closure. As the next statement to be generated is submitted at the time of current statement generation, this statement request needs to be removed when there is a change in frequency. <p>This method will check for statement request that are to be executed only on or after the current processing date</p> Note: This method will not remove the corresponding action of the statements request. It is assumed that all the actions for the deposit that is getting closed will be removed before calling this. @param accountId @param depositNumber
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Statement request once given by the customer or the system, will log an action for it generation. When the action is take up for execution, the executor will need to know all the statement requests that has to be generated on the current process date. This method will fetch all such statement requests that will be later generated. This method will fetch all the {@link TDStatementRequest} that needed to be generated on the given date. @param statementGenerationDate @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will fetch all the deposit details for the given accountId. @param accountId @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will fetch the deposit details for the given account id and deposit number. @param accountId @param depositNumber @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		Statement request once given by the customer or the system, will log an action for it generation. When the action is take up for execution, the executor will need to know all the statement requests that has to be generated on the current process date. This method will fetch all such statement requests that will be later generated. This method will fetch all the {@link TDStatementRequest} that needed to be generated on the given date. @param processDate @param nextProcessDate @param accountId @param depositNumber @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to delete a deposit statement request. Statement requests may need to be deleted if for system initiated cases, there is a change in the statement frequency or account closure. As the next statement to be generated is submitted at the time of current statement generation, this statement request needs to be removed when there is a change in frequency.<p>This method will check for statement request that are to be executed only on or after the current processing date </p> Note: This method will not remove the corresponding action of the statements request. It is assumed that all the actions for the deposit that is getting closed will be removed before calling this. @param accountId @param depositNumber
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will perform multiple cheque deposits along with cash deposit.  The MultiFundDepositApplication  <li>calls  {@link OutgoingClearingService#save} to validate details of each cheque before deposit.   </li><li>calls  {@link com.ofss.fc.app.dda.service.transaction.MultiFundDepositApplication#depositMultiFund(com.ofss.fc.app.dda.dto.transaction.MultiFundDepositDTO)} to deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>multiple cheques along with cash in the beneficiary account. This will return corresponding accounting amount tag details with credit leg information for each cheque and cash deposit processed in the transaction.</p> <p><code>&lt;br /&gt; &lt;/li&gt;&lt;li&gt;calls&lt;br/&gt; {@link com.ofss.fc.app.account.multifunddeposit.service.MultiFundDepositApplication#raiseAccounting(ApplicationContext, MultiFundDepositDTO, String, TransactionContext, FundsExchangeResponse, MultiFundDepositResponse)} to raise accounting for each cheque and cash deposit of multi fund deposit transaction. This will generate accounting amount tag details with debit leg information before raising the accounting. &lt;br /&gt; &lt;/li&gt;&lt;li&gt;calls&lt;br/&gt; {@link MultiFundDepositApplication#raiseAlert(MultiFundDepositDTO)} to send alert to the customer once Multi Fund deposit transaction is done successfully.&lt;/li&gt;&lt;br /&gt; @fcb.param MI, ApplicationContext Indicates basic information about the interaction being initiated with the system. &lt;li&gt; {@link ApplicationContext}&lt;/li&gt; @fcb.param MI, MultiFundDepositDTO Indicates all the information related to multi fund deposit(cashTransactionDTO,instrumentTransferCaptureDTO,wrappedPriceFeatureDTO) &lt;li&gt; {@link MultiFundDepositDTO}&lt;/li&gt; @return TransactionStatus &lt;li&gt;{@link com.ofss.fc.service.response.TransactionStatus}&lt;/li&gt; @throws FatalException</code></p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method fetches the total transaction amount of a multi fund deposit transaction. @fcb.param MI, MultiFundDepositResponse, response <li> {@link com.ofss.fc.app.dda.dto.transaction.MultiFundDepositResponse}</li> @return totalAmount, This is the total transaction amount in a multi fund deposit transaction.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method fetches the total transaction amount of a multi fund deposit transaction. @fcb.param MI, MultiFundDepositDTO, multiFundDepositDTO <li> {@link com.ofss.fc.app.account.dto.multifunddeposit.MultiFundDepositDTO}</li> @return totalAmount, This is the total transaction amount in a multi fund deposit transaction.
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This will generate the Multi Fund Deposit Data Transfer Object required for the DDA module to process fund transfer to the provided CASA account. @fcb.param MI, MultiFundDepositDTO, multiFundDepositDTO <li> {@link com.ofss.fc.app.account.dto.multifunddeposit.MultiFundDepositDTO}</li> @fcb.param MI, FundsExchangeResponse, fundsExchangeResponse <li> {@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @return MultiFundDepositDTO <li> {@link com.ofss.fc.app.dda.dto.transaction.MultiFundDepositDTO}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method used to log <a href="#">{@link DDActionRequest}</a>.It takes <a href="#">{@link IActionType}</a> as input instead of <a href="#">{@link DDActionType}</a>. It's purpose is to enable extending the action types for demand deposit.This method will perform following activities</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Validate all input using <code>checkManadatoryInput()</code></li> <li>&lt;li&gt;Validate the extraAttribute string to make sure it doesn't contain invalid characters.</li> <li>&lt;li&gt; Call <code>fetchPickupDate</code> to get pickupDate</li> <li>&lt;li&gt;If the pickupDate and the dueDate are not the same and holidayEpochType is not null in additionalActionData, then use the holidayEpochType instead of epochType for logging the action. Else continue using epochType.</li> <li>&lt;li&gt;Load DDAction from DDActionRepository using input.actionType.</li> <li>&lt;li&gt;Create instance of DDActionRequest and do instance.setAction(ddAction)</li> <li>&lt;li&gt;If <code>input.epochType == EndOfDay</code></li> <li>&lt;li&gt;Load <a href="#">{@link DDEODActionSet}</a> for given account and action date using <a href="#">{@link ActionRepository#load}</a></li> <li>&lt;li&gt; If <a href="#">{@link DDEODActionSet}</a> is not present create it using <a href="#">{@link ActionSetFactory#createDDAccountBasedActionSet}</a>,call <a href="#">{@link DDEODActionSet#addAction}</a></li> <li>&lt;li&gt; <a href="#">{@link ActionRepository#insert}</a>.</li> <li>&lt;li&gt;If <a href="#">{@link DDEODActionSet}</a> is present ,call <a href="#">{@link DDEODActionSet#addAction}</a></li> <li>&lt;li&gt; <a href="#">{@link ActionRepository#update}</a>.</li> <li>&lt;/ol&gt;</li> <li>&lt;li&gt;else If <code>input.epochType == BeginningOfDay</code></li> <li>&lt;li&gt;Load <a href="#">{@link DDBODActionSet}</a> for given account and action date using <a href="#">{@link ActionRepository#load}</a></li> <li>&lt;li&gt; If <a href="#">{@link DDBODActionSet}</a> is not present create it ,call <a href="#">{@link DDBODActionSet#addAction}</a></li> <li>&lt;li&gt; Call <code>fetchPickupDate</code> to get pickupDate</li> <li>&lt;li&gt; <a href="#">{@link ActionRepository#insert}</a>.</li> <li>&lt;li&gt;If <a href="#">{@link</a></li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DDBODActionSet} is present ,call {@link DDBODActionSet#addAction} &lt;br&gt; {@link ActionRepository#update}. &lt;/ol&gt; &lt;p&gt; When an existing action is logged, the extraAttribute string will also be compared. Only if the extraAttributes are also the same, no update will be done. Else the existing action will be removed and new action with the new extraAttributes will be logged instead. &lt;/p&gt; @param partyId,primary account holder @param accountId, demand deposit account id . @param ddActionType, action to log on demand deposit account. @param dueDate, action execution date. It can not be backdated.It should be greater than or equal to bank process date. @param holidayTreatment, Holiday treatment method if due date falls on holiday.This must be left as null if possible so that seed will be used. @param epochType,batch process like end of the day ,beginning of the day. @param additionalActionData,additional data required for logging action</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method used to remove {@link DDAction} from {@link DDActionSet}.It takes {@link IActionType} as input instead of {@link DDActionType}. It's purpose is to enable extending the action types for demand deposit actions. This method will perform following activities &lt;ol&gt; &lt;li&gt;Validate all input using {@code checkManadatoryInput()} &lt;li&gt; Load DDAction from DDActionRepository using @link input.ddActionType}. &lt;li&gt;Create instance of DDActionRequest and do instance.setAction(ddAction) &lt;li&gt;If {@code input.epochType == EndOfDay} &lt;ol&gt; &lt;li&gt;Load {@link DDEODActionSet} for given account and action date using {@link ActionSetRepository#loadDDAction} &lt;li&gt;If {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DDEODActionSet} is not present return &lt;li&gt;If {@link DDEODActionSet} is present ,call {@link DDEODActionSet#removeAction} &lt;br&gt; {@link ActionRepository#update}. &lt;/ol&gt; &lt;li&gt;If {@code input.epochType == BeginningOfDay} &lt;ol&gt; &lt;li&gt;Load {@link DDBODActionSet} for given account and action date using {@link ActionSetRepository#loadDDAction} &lt;li&gt;If {@link DDBODActionSet} is not present return &lt;li&gt;If {@link DDBODActionSet} is present ,call {@link DDBODActionSet#removeAction} &lt;br&gt; {@link ActionRepository#update}. &lt;/ol&gt; &lt;/ol&gt; &lt;p&gt; If there has been a change in calendar(see {@link #fetchIsCalChangedFalg()}), then remove action can no longer be done based on the pickup as the new pickup dates may not be the same as the old ones. Hence a like query will be fired based on the dueDate and the action will be removed from the actionsets thus fetched. see {@link ActionSetRepository#fetchAllDDEODActionSet()}, {@link ActionSetRepository#fetchAllDDTODActionSet()} &lt;/p&gt; @param accountId, demand deposit account id . @param actionType, action to remove on demand deposit account. @param dueDate, action execution date. It can not be back dated.It should be greater then or equal to bank process date. @param holidayTreatment, Holiday treatment method if due date falls on holiday. @param epochType, batch process like end of the day ,beginning of the day. @param additionalActionData, additional data that may be required to remove the action.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<p>This method fetches all the deposits for which the deposits have been maintained and do not have their status as closed and their respective deposit details. Following are the steps to be followed in this method:-</p> <ol style="list-style-type: none"> <li>Create an instance of AccountNominationsResponse</li> <li>Fetch the transaction status using fetchTransactionStatus().</li> <li>Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments</li> <li>Check for the existence of the account by calling fetchAccount() of TermDepositAccountApplicationService through the TD adapter.</li> <li>Create an instance of Domain Service and invoke it's fetchAllNominatedDeposits() method by passing accountID. This would return an array of depositNumbers.</li> <li>For each deposit in the array fetched, call fetchDeposit() of DepositApplicationService through the TD adapter.</li> <li>Set each response in the Inquiry Response if the deposit status is not closed.</li> <li>The inquiry response will not have any party details.</li> <li>Fill the Transaction Status by calling the method fillTransactionStatus and pass the transaction status as an argument to it.</li> <li>Set the response status as transaction status</li> <li>Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks.</li> <li>Close the interaction in finally block.</li> <li>return the Response of this operation.</li> </ol> <p>/ @param sessionContext @param accountId @param accountType @return AccountNominationsResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		@param accountNominationRequestDTO @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method fetches the product code for the input account Id. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to populate a DepositsFeeRequestInfo object. @param accountId @param pricingTicketNumber @param feeDetail @param accountDTO @return depositsFeeRequestInfo
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method is used to create new deposit based action set for term deposit account based on {@link EpochType}.This method will perform following activities <ol style="list-style-type: none"> <li>Create object reference <code>DepositBasedActionSet depositBasedActionSet</code> <li>If input.epochType = EndOfDay then   depositBasedActionSet = new TDEODActionSet(); <li>else input.epochType = BeginningOfDay then   depositBasedActionSet = new TDBODActionSet(); <li> <code>depositBasedActionSet.setAccountId(input.accountId)</code> <code>depositBasedActionSet.setDepositId(input.depositId)</code> <li>return depositBasedActionSet; <li> @param accountId @param depositId</li> </li></li></li></li></li></ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		This method will be used to post the fees pending for the account and update the account balance.   @param accountId @param pricingMap @throws FatalException @throws FatalException
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<ol>The method populates the DepositsFeeRequestInfo which is then passed to the FeesManagementService as input. The FeeManagementService requires instance of DepositsFeeRequestInfo </ol> <ol> <li>set DepositsFeeRequestInfo.ModuleType = TaskModuleType.CASA</li> <li>load account by {@link fetchDemandDepositAccount(accountId: String, moduleType: ModuleType): IAccount} <li>set depositsFeeRequestInfo.partyId = account.getPartyId(</li> <li>set depositsFeeRequestInfo.accountId = accountTransaction.getAccountId</li> <li>set depositsFeeRequestInfo.eventId = accountTransaction.getEventId(</li> <li>set depositsFeeRequestInfo.currencyCode[0] = accountTransaction.getCurrency(</li> <li>set depositsFeeRequestInfo.PricingDetail = {@link getPricingDetail(FeeDetails: feeDetails): PricingDetails pricingDetails}</li> <li>set depositsFeeRequestInfo.EvaluationDate = AccountTransaction.TransactionDates.PostingDate</li> <li>set depositsFeeRequestInfo.userReferenceNumber = accountTransaction.UserReferenceNumber</li> <li>set depositsFeeRequestInfo.TransactionAmount = accountTransaction.TransactionAmount.AmountInAccountCurrency</li> </ol> @param accountTransaction @param moduleType @param feeAmounts @param priceTicketNumber @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				DepositsFeeRequestInfo
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		@param priceTicketNumber @param feeAmount @param accountDebitResponse @param financialEventType @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		<ol>The method populates the DepositsFeeRequestInfo which is then passed to the FeesManagementService as input. The FeeManagementService requires instance of DepositsFeeRequestInfo </ol> <ol>The method set the TaskModuleType as follows: <li>if the moduleType == DDA then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.CASA</li> <li>if the moduleType == TD then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.TERM_DEPOSIT</li> <li>load account by {@link fetchDemandDepositAccount(accountId: String, moduleType: ModuleType): IAccount} <li>set depositsFeeRequestInfo.partyId = account.getPartyId()</li> <li>set depositsFeeRequestInfo.accountId = accountTransaction.getAccountId</li> <li>set



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> depositsFeeRequestInfo.eventId = accountTransaction.getEventId()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.currencyCode[0] = accountTransaction.getCurrency()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.PricingDetail = {@link getPricingDetail(FeeDetails: feeDetails): PricingDetails pricingDetails}&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.EvaluationDate = AccountTransaction.TransactionDates.PostingDate&lt;/li&gt; &gt; &lt;li&gt;set depositsFeeRequestInfo.userReferenceNumber = accountTransaction.UserReferenceNumber&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.TransactionAmount = accountTransaction.TransactionAmount.AmountInAc countCurrency&lt;/li&gt; &lt;/ol&gt; @param accountTransaction @param moduleType @param feeAmounts @param priceTicketNumber @return DepositsFeeRequestInfo </pre>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deposit		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitAccount		<p>This method will debit the savings account with provided amount. If cheque no is provided, then the account's cheque will be marked as paid. It should also generate a entry in account audit trail @param nsfTreatmentType Indicates the manner of debit. It can be used to forcefully debit the account under force post conditions @param context @param accountID @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitAccount		This will raise accounting in case of direct credit to CASA during multi fund deposit transaction. @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, FundsExchangeResponse, fundsExchangeResponse <li> {@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @fcb.param MI, String, accountId @fcb.param MI, FundDepositResponse, fundDepositResponse <li> {@link com.ofss.fc.app.dda.dto.transaction.FundDepositResponse}</li> @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addStatementItem	The method will add the given statement item to the transient Map {@code #statementItemMap}. @param transactionReferenceNumber @param subSequenceNumber TODO @param accountId TODO	This method takes care of all the consolidation details that is required and returns the inquiry response object filled with the statement items inquired.It follows the following steps: <ol> <li>Initialize sortedDDStatementItemDTOSet:TreeSet<DDStatementItemDTO> using the postingDateComparator</li> <li>Initialize a {@link StatementStatistics} object statementStatistics, and {@link DDStatementItemConsolidator} object dDStatementItemConsolidator</li> <li>Fetch the isChequeConsolidationRequired and isInteresAndTaxConsolidationRequired using the account product adaptor.</li> <li>Loop thorough all the financial statement items in {@link StatementInquiryResponse}.Do the following: <ol> <li>Fetch the consolidation type of the current item.</li> <li>If the type equals ConsolidationType.NONE, Assemble the item,add it to the sortedDDStatementItemDTOSet and update statementStatistics. <li>If the type equals ConsolidationType.BULKFEE, add it to the dDStatementItemConsolidator</li> <li>If the type

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>equals ConsolidationType.CHEQUE and isChequeConsolidationRequired, add it to dDStatementItemConsolidator&lt;/li&gt; &lt;li&gt;If the type equals ConsolidationType.INTEREST_AND_TAX and isInteresAndTaxConsolidationRequired, add it to dDStatementItemConsolidator&lt;/li&gt; &lt;/ol&gt; &lt;li&gt;Call the {@link DDStatementItemConsolidator#fetchStatementItems} and add the result list to sortedDDStatementItemDTOSet.&lt;/li&gt; &lt;li&gt;Loop through all the all financial statement items in statementInquiryResponse and add it to sortedDDStatementItemDTOSet after assembling them.&lt;/li&gt; &lt;li&gt;Convert the TreeSet into {@link DDStatementItemDTO}[] and set it in the DDTransactionInquiryResponse along with statementStaticsics and return the same.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param ddTransactionInquiryResponseDTO @param statementInquiryResponse @param postingDateComparator @return {@link DDTransactionInquiryResponse} @throws FatalException</p>
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addStatementItem	<p>The method will add the given statement item to the transient Map {@code #statementItemMap}. @param transactionReferenceNumber @param subSequenceNumber TODO @param accountId TODO</p>	<p>This method takes care of all the consolidation details that is required and returns the inquiry response object filled with the statement items inquired.It follows the following steps: &lt;ol&gt; &lt;li&gt;Initialize sortedTDStatementItemDTOSet:TreeSet&lt;TDStatementItemDTO&gt; using the postingDateComparator&lt;/li&gt; &lt;li&gt;Initialize a {@link StatementStatistics} object statementStatistics, and {@link TDStatementItemConsolidator} object tDStatementItemConsolidator&lt;/li&gt; &lt;li&gt;Fetch the isChequeConsolidationRequired and isInteresAndTaxConsolidationRequired using the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>account product adaptor.</p> <ul style="list-style-type: none"> <li>Loop thorough all the financial statement items in {@link StatementInquiryResponse}.Do the following: <ul style="list-style-type: none"> <li>Fetch the consolidation type of the current item.</li> <li>If the type equals ConsolidationType.NONE, Assemble the item,add it to the sortedDDStatementItemDTOSet and update statementStatistics.</li> <li>If the type equals ConsolidationType.BULKFEE, add it to the dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.CHEQUE and isChequeConsolidationRequired, add it to dDStatementItemConsolidator</li> <li>If the type equals ConsolidationType.INTEREST_AND_TAX and isInteresAndTaxConsolidationRequired, add it to dDStatementItemConsolidator</li> </ul> </li> <li>Call the {@link TDStatementItemConsolidator#fetchStatementItems} and add the result list to sortedTDStatementItemDTOSet.</li> <li>Loop through all the all financial statement items in statementInquiryResponse and add it to sortedDDStatementItemDTOSet after assembling them.</li> <li>Convert the TreeSet into {@link TDStatementItemDTO}[] and set it in the DDTransactionInquiryResponse along with statementStaticsics and return the same.</li> </ul> <p>@param sessionContext @param tdTransactionInquiryResponseDTO @param statementInquiryResponse @param postingDateComparator @return TDTransactionInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	addStatementItem	The method will add the given statement item to the transient Map {@code #statementItemMap}. @param transactionReferenceNumber @param subSequenceNumber TODO @param accountId TODO	Financial statement item will be logged for demand deposit account on stimulus to financial event which generally changes {@link DemandDepositBalances}.like cash deposit,cash withdrawal ,fund transfer,cheque deposit etc. This method is used to create new statement item. @param ddFinStmntItem,financial statement item for demand deposit account. @param ddFinStmntItem
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getFinancialStatementItems	The method will return all statement items that have been logged with the given reference number in the current session. If none such statements have been logged then it will return null. @param referenceNumber @param accountId TODO @return	This method is used to modify {@link SystemDefinedStatus} of the demand deposit account.This method also does auditing activity. This method will perform activity is following sequence. <ol> <li>oldSystemDefinedStatus = this.systemDefinedStatus. <li>if oldSystemDefinedStatus equal to input.systemDefinedStatus then return <li>this.systemDefinedStatus = input.systemDefinedStatus. <li>Create {@link SystemDefinedStatusAuditItem} with following specification <ol> <li>accountId = this.key.accountId <li>fromEffectiveDate = oldSystemDefinedStatus.effectiveDate <li>toEffectiveDate = input.effective date. <li>fromProcessDate = oldSystemDefinedStatus.processDate <li>toProcessDate = input.processDate. <li>systemDefinedStatus = oldSystemDefinedStatus <li>user number = get it from thread local->Session context. </ol> @param newStatus @param effectiveDate @param isInterestIntimationRequired

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getFinancialStatementItems	The method will return all statement items that have been logged with the given reference number in the current session. If none such statements have been logged then it will return null. @param referenceNumber @param accountId TODO @return	Financial statement item will be logged for demand deposit account on stimulus to financial event which generally changes {@link DemandDepositBalances}.like cash deposit,cash withdrawal ,fund transfer,cheque deposit etc. This method is used to create new statement item. @param ddFinStmntItem,financial statement item for demand deposit account. @param ddFinStmntItem
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPseudoStatementEntries	This method is to log the statement entries.This method is added, since the {@link #prepareDDFinancialStatementItem (AccountTransaction)} is a private method, which contains the actual logic of adding a statement item (Financial). @param isBalChangeReq TODO @param accountTransaction	At the time of Online / periodic fee charging, system will check whether account has any alternate account for fees.  If account has alternate account for fees, it will check whether   ΓÇÿPseudo Entries for fees required in parent accountΓÇÖ parameter is checked for the account.  If checkbox is checked, in the statement inquiry screen / ad-hoc account statement for parent account,  system will display pseudo entries for redirected fees. @param debitRequestDTO @param alternateAccountId @param accountTransaction @param accountTransactionAdapter
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logPseudoStatementEntries	This method is to log the statement entries.This method is added, since the {@link #prepareDDFinancialStatementItem (AccountTransaction)} is a private method, which contains the actual logic of adding a statement item (Financial). @param isBalChangeReq TODO @param accountTransaction	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isPseudoStatementEntryRequiredForFee		At the time of Online / periodic fee charging, system will check whether account has any alternate account for fees.  If account has alternate account for fees, it will check whether   ΓÇÿPseudo Entries for fees required in parent accountΓÇÖ parameter is checked for the account.  If checkbox is checked, in the statement inquiry screen / ad-hoc account statement for parent account,  system will display pseudo entries for redirected fees. @param debitRequestDTO @param alternateAccountId @param accountTransaction @param accountTransactionAdapter
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isPseudoStatementEntryRequiredForFee		@param primaryAccountId @param alternateAccountId @param accountTransaction @param accountTransactionAdapter @param parentAccountNarration
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	applyHardShipRelief	To approve the DDA hardship relief and apply the relief on account in hardship @param applicationContext @param applicationId @throws FatalException	To approve the DDA hardship relief and apply the relief on account in hardship @param applicationContext @param applicationId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	applyHardShipRelief	To approve the DDA hardship relief and apply the relief on account in hardship @param applicationContext @param applicationId @throws FatalException	This method will mark the hardship reliefs and request APPROVED and apply the STP reliefs. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkIfCollectedBalanceAccount	<p>This method will load the CASA account and with the product code of the account, it will fetch whether the account is in book balance or collected balance. </p> @param accountId @return	This method raises accounting in case of Cheque Deposit to CASA during a Multi Fund Deposit transaction. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, TransactionLogDTO, transactionLog <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO }</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO, accountingEventAmountTagDetailsDTOArray <li>{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, MessageLogItemDTO, messageLogItem <li>{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}</li> @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkIfCollectedBalanceAccount	<p>This method will load the CASA account and with the product code of the account, it will fetch whether the account is in book balance or collected balance. </p> @param accountId @return	This method populates the Clearing Transaction Data Transfer Object. This data transfer object holds the information of all the transactions in which the unclear funds are modified during the MFD transaction. @fcb.param MI, String, accountId @fcb.param MI, InstrumentLogItemDTO, instrumentDetail <li>{@link com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO}</li> @fcb.param MI, TransactionLogDTO, transactionLogDTO <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO}</li> @return ClearingTransactionDTO <li>{@link com.ofss.fc.app.dda.dto.account.cheque.ClearingTransactionDTO}</li>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchEffectiveBackdateOnAccount	This method will check if the account has any backdated transaction for the current posting date. @param accountId @return	Method validateBackdatedTransactionsOnAccount is called to check whether there is any backdated transactions occurred on the account for the given posting date.   Since R2.2</br>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This is a private method to fetch the bank name from the bank code. Bank name needs to be printed at the end of each message. @param sessionContext @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is minor and the operating instruction channel access type is no for given channel then return true,false otherwise @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will fetch the bank code for given account ID @param partyAccountRelationshipDetailsDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is operationally major and the operating instruction channel access type is no for given channel then return true,false otherwise @param partyAccountRelationshipDetailsDTO @return
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	deriveLeadLagConfiguration		This method based on account type calls to load the lead lag configuration from appropriate adapter. @param eventId @param accountType @param accountId
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkProportionateODForGivenAccountAndDate	This method will check if there is any proportionate od for given account and date,if there is any it will check that the due date is less than effective date ,if true than account branch transfer is not allowed,allowed in case of false.	Method validateBackdatedProportionateODOnAccount is called to check whether there is any proportionate od balance change to occur on the account for the given posting date.   Since R2.2</br>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	checkDebitInstructionGivenAccount	This method will check if there is any debit instruction which is Uninitiated for given account and an effective date ,if true than account branch transfer is not allowed,allowed in case of false.	Method validateDebitInstructionsOnAccount is called to check whether there is any debit instructions to occur on the account for the given posting date.   Since R2.2</br>
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isBodRunForToday	This will check whether bod has run for today in this environment.If the batch mode is on then it won't check because batch mode is on means batch is already running.This requires in case of online transaction which doesn't know batch is running or not. @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isBodRunForToday	This will check whether bod has run for today in this environment.If the batch mode is on then it won't check because batch mode is on means batch is already running.This requires in case of online transaction which doesn't know batch is running or not. @return	This will check whether bod has run for today in this environment.If the batch mode is on then it won't check because batch mode is on means batch is already running.This requires in case of online transaction which doesn't know batch is running or not. @return
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isBodRunForToday	This will check whether bod has run for today in this environment.If the batch mode is on then it won't check because batch mode is on means batch is already running.This requires in case of online transaction which doesn't know batch is running or not. @return	Method validateCutoff is called to check whether cutoff has been executed for the day or not. No account transfer request after the EOD for the day.
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	debitBackdatedFees	/ <ol> <li>In case of backdated account opening, backdated fees are applicable. <li>Typically in case of OD settlement execution , we charge fees with backdated value date. @param accountTransactionReturn @return @throws FatalException	
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	handleConsolidatedFees		
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	handleConsolidatedFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	handleConsolidatedFees		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logReversalPseudoStatementEntries		@param primaryAccountId @param alternateAccountId @param accountTransaction @param accountTransactionAdapter @param parentAccountNarration
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logReversalPseudoStatementEntries		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logFinancialRequest	This method calls to log request of CASA financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param transactionAmounts @param accountId	{@link IDDARequestResponseLogService#logFinancialRequest(String[], Map, Object[], TransactionDatesDTO, TransactionAmountsDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logFinancialRequest	This method calls to log request of CASA financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param transactionAmounts @param accountId	{@link IAccountDDAAdapter#logFinancialRequest(String[], Map, Object[], TransactionDatesDTO, TransactionAmountsDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logFinancialResponse	This method calls to log response of DDA financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param transactionAmounts @param	{@link IDDARequestResponseLogService#logFinancialResponse(String[], Map, Object[], TransactionDatesDTO, TransactionAmountsDTO, String)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			accountId	
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logFinancialResponse	This method calls to log response of DDA financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param transactionAmounts @param accountId	{@link IAccountDDAAdapter#logFinancialResponse(String[], Map, Object[], TransactionDatesDTO, TransactionAmountsDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logNonFinancialRequest	This method calls to log request of DDA non financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param accountId	{@link IDDARequestResponseLogService#logNonFinancialRequest(String[], Map, Object[], TransactionDatesDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logNonFinancialRequest	This method calls to log request of DDA non financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param accountId	{@link IAccountDDAAdapter#logNonFinancialRequest(String[], Map, Object[], TransactionDatesDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logNonFinancialResponse	This method calls to log response of DDA non financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param accountId	{@link IDDARequestResponseLogService#logNonFinancialResponse(String[], Map, Object[], TransactionDatesDTO, String)}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	logNonFinancialResponse	This method calls to log response of DDA non financial transaction. @param params @param paramMap @param affectedEntity @param transactionDates @param accountId	{@link IAccountDDAAdapter#logNonFinancialResponse(String[], Map, Object[], TransactionDatesDTO, String)}
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	clearCacheInWarningIndicatorReferenceFactor	This method will clear the cache used in case of warning indicators.	This represents the updateWarningIndicator method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,WarningIndicatorDTO,warningIndicatorDTO, ,warningIndicatorDTO TODO: Explain the warningIndicatorDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getBasicAccountInformation		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountDDAAdapter	doContributionProcess		This method will process the disbursement instructions for the settlement payout instruction.  It will interact with Loan module for the processing of the disbursement instruction via  AccountLoanAdapter.  @param applicationContext @param instructionId @paramactionDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	doContributionProcess		This method will process the contribution instructions for the settlement payout instruction.  It will interact with CASA module for the processing of the contribution instruction via  AccountDDAAdapter.  @param applicationContext @param instructionId @paramactionDate @throws FatalException
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	doSettlementPayoutReversal		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isOfficeAccount		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isOfficeAccount		
IAccountDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	isOfficeAccount		This method used to log {@link DDActionRequest}.This method will perform following activities <ol> <li>Validate all input using {@code checkManadatoryInput()}  Call {@code fetchPickupDate} to get pickupDate <li>If the pickupDate and the dueDate are not the same and holidayEpochType is not null in additionalActionData, then use the holidayEpochType instead of epochType for logging the action. Else continue using epochType. <li>Load DDAction from DDActionRepository using input.actionType. <li>Create instance of DDActionRequest and do instance.setAction(ddAction) <li>If {@code

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> input.epochType == EndOfDay} &lt;ol&gt; &lt;li&gt;Load {@link DDEODActionSet} for given account and action date using {@link ActionRepository#load} &lt;li&gt;If {@link DDEODActionSet} is not present create it using {@link ActionSetFactory#createDDAccountBasedActionSet} ,call {@link DDEODActionSet#addAction}&lt;br&gt; {@link ActionRepository#insert}. &lt;li&gt;If {@link DDEODActionSet} is present ,call {@link DDEODActionSet#addAction} &lt;br&gt; {@link ActionRepository#update}. &lt;/ol&gt; &lt;li&gt;else If {@code input.epochType == BeginningOfDay} &lt;ol&gt; &lt;li&gt;Load {@link DDBODActionSet} for given account and action date using {@link ActionRepository#load} &lt;li&gt;If {@link DDBODActionSet} is not present create it ,call {@link DDBODActionSet#addAction} &lt;br&gt; Call {@code fetchPickupDate} to get pickupDate &lt;br&gt; {@link ActionRepository#insert}. &lt;li&gt;If {@link DDBODActionSet} is present ,call {@link DDBODActionSet#addAction} &lt;br&gt; {@link ActionRepository#update}. &lt;/ol&gt; @param partyId,primary account holder @param accountId, demand deposit account id . @param ddActionType, action to log on demand deposit account. @param dueDate, action execution date. It can not be backdated.It should be greater than or equal to bank process date. @param holidayTreatment, Holiday treatment method if due date falls on holiday. @param epochType,batch process like end of the day ,beginning of the day. @param additionalActionData,additional data required for logging action </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateContributionInstructionStatus		This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param applicationContext @param settlementPayoutInstruction
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	updateContributionInstructionStatus		This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param settlementPayoutInstruction
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountTitle		This method does the validations if there is a change in Sanctioned Amount/Term. @param sessionContext @param loanAccountAmendConfigurationDTO @param amendTypes
IAccountDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountDDAAdapter	getAccountTitle		
ILimitsAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountingAdapter	raiseAccountingEvent	This method calls the accounting module to raise the accounting   entries for the deferred fees processing @param facilitySettlementAccountingEventDTO @return TransactionStatus @throws BusinessException	This method calls passes the accounting entry details to the limits accounting adapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountingAdapter	raiseAccountingEvent	This method calls the accounting module to raise the accounting   entries for the deferred fees processing @param facilitySettlementAccountingEventDTO @return TransactionStatus @throws BusinessException	This method calls the accounting module to raise the accounting   entries for the deferred fees processing @param facilitySettlementAccountingEventDTO @return TransactionStatus @throws BusinessException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateOfferDocumentStatus	This method updates the execution Status of the generated document. The document is generated by DMS and the ID will be stored in Origination module. During the Acceptance and Execution of Documents, the status of the document will be updated. updateOfferDocumentStatus method will call the corresponding services in DMS to update the document. @param sessionContext @param documentId @param status @throws FatalException	This method is used to update offer document status for the application. The flow of the method is as follows,  <ul> OfferDocumentLinkageDTOBuilder <li></li> <li>Iterate through the input Offer documents and convert to OfferDocumentLinkageDTO using the builder @link{OfferDocumentLinkageDTOBuilder </li> <li>Set the OfferDocumentLinkageDTO and the checklist information to LendingOfferChecklistReferenceDTO</li> <li>Update the status by invoking the service @link{com.ofss.fc.app.origination.service.lending.core.offer.LendingOfferApplicationService .updateOfferDocumentStatus } </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,@link{com.ofss.fc.app.origination.dto.lending.core.application.OfferDocumentDTO}, an array of these will contain information on offer documents such as applicationId, documentId, checklistId, etc @return @link{com.ofss.fc.service.response.TransactionStatus}, This will represent whether the update was done successfully @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateOfferDocumentStatus	This method updates the execution Status of the generated document. The document is generated by DMS and the ID will be stored in Origination module. During the Acceptance and Execution of Documents, the status of the document will be updated. updateOfferDocumentStatus method will call the corresponding services in DMS to update the document. @param sessionContext @param documentId @param status @throws FatalException	This method 1. updates the Status, expiry Date, reason code if any, and action date of the offer documents in Origination 2. Creates a record in the history table in case the status of a document has been updated. 3. Updates the corresponding document details in DMS for eg: the execution flag, executed date etc. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingApplicationOfferDocumentReferenceDTO @return TransactionStatus
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateOfferDocumentStatus	This method updates the execution Status of the generated document. The document is generated by DMS and the ID will be stored in Origination module. During the Acceptance and Execution of Documents, the status of the offer document will be updated. updateOfferDocumentStatus method will call the corresponding services in DMS to update the document. @param sessionContext @param documentId @param status @throws FatalException	This method updates the execution Status of an offer document. The document is generated and stored in DMS and the corresponding ID is stored in Origination module. During the Acceptance and Execution of Documents, the status of the offer document will be updated. updateOfferDocumentStatus method will call the corresponding service in DMS to update the execution status of the document. @param sessionContext @param OutboundDocumentListRecordDTO @param status @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	inquireOfferDocumentForChecklistId	This method retrieve the generated offer document for an application. The document is generated by DMS and the ID will be stored in Origination module. Origination will give the APPLICATION as the entity type and the checklistId as the value. DMS will return the list of generated documents. @param sessionContext @param checklistId @return @throws FatalException	The method is used to fetch all the the offer documents that have been dispatched to the customer and are waiting to be accepted / executed. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @return OfferDocumentLinkageResponse @exception   {@link OriginationErrorConstants#CHECKLIST_NOT_FOUND} : FC_OR_OF_0043   This error occurs when:  The offer documents are retrieved for an application for which the offer document checklist has not been created
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	inquireOfferDocumentForChecklistId	This method retrieve the generated offer document for an application. The document is generated by DMS and the ID will be stored in Origination module. Origination will give the APPLICATION as the entity type and the checklistId as the value. DMS will return the list of generated documents. @param sessionContext @param checklistId @return @throws FatalException	The method is used to fetch the history of all the offer documents for the given application ID @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @return OfferDocumentLinkageHistoryResponse @exception   {@link OriginationErrorConstants#OFFER_HISTORY_NOT_FOUND} : FC_OR_OF_0048    {@link OriginationErrorConstants#OFFER_HISTORY_NOT_FOUND} : FC_OR_OF_0048   This error occurs when there are no history offer records for the given

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				application
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	inquireOfferDocumentForChecklistId	This method retrieve the generated offer document for an application. The document is generated by DMS and the ID will be stored in Origination module. Origination will give the APPLICATION as the entity type and the checklistId as the value. DMS will return the list of generated documents. @param sessionContext @param checklistId @return @throws FatalException	This method will be called only when all the offer documents are dispatched. It will construct the OfferDocumentLinkageDTO array. @param sessionContext @param checkListId @param offerDocReferenceId @param offerDocReferenceType @return OfferDocumentLinkageResponse @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	inquireOfferDocumentForChecklistId	This method retrieve the generated offer document for an application. The document is generated by DMS and the ID will be stored in Origination module. Origination will give the APPLICATION as the entity type and the checklistId as the value. DMS will return the list of generated documents. @param sessionContext @param checklistId @return @throws FatalException	This method calls publishOriginationActivity from OriginationEventApplicationService which logs and processes events related to offer letter and offer documents.It logs event depending upon status of Offer Letter and Offer Documents.It will internally call registerActivity from AlertsNotificationHandler which performs eventProcessing. @param sessionContext @param offerLetter @param submissionId @param applicationId @param offerDocumentLinkagesSet @param checklistId
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	inquireOfferDocumentForChecklistId	This method retrieve the generated offer document for an application. The document is generated by DMS and the ID will be stored in Origination module. Origination will give the APPLICATION as the entity type and the checklistId as the value. DMS will return the list of generated documents. @param sessionContext @param checklistId @return @throws FatalException	This method retrieves the generated offer document(s) for an application. The document are generated by DMS and the corresponding ID's are stored in Origination module. @param sessionContext @param checklistId @return OutboundDocumentListRecordDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	createDocumentChecklist	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method triggers the creation of offer document checklist for the application. The following steps are performed: 1. If offer documents checklist has already been created for the application, go to step 6 2. Pre create offer validations are performed. 3. If all the validations are successful, the facts required for document generation are collected and passed to DMS 4. Based on the facts, a checklist with a list of documents is created by the DMS module by executing document policies 5. The list of documents along with the checklist ID is returned to origination. 6. The checklist Id is stored in origination for future reference and the list of documents obtained in the DMS response is returned to the UI for generation and dispatch @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @throws FatalException @exception   {@link OriginationErrorConstants#OFFER_GENERATION_POLICY_NA} : FC_OR_OF_0003   This error occurs when:  The DMS returns null response after executing policy    {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057   This error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.    {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084   This error occurs when:  Warning indicators which prevent account opening are present for applicants applying to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#COLLATERAL_EVALUATION_NOT_DONE} : FC_OR_OF_0010 &lt;br&gt; This error occurs when:&lt;br&gt; Collateral evaluation (valuation and title search) have not been completed for a collateral(s), if any. An overridable policy exception is thrown as an information message is displayed to the customer. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040 &lt;br&gt; This error occurs when any of the validations performed before creating checklist, fail&lt;br&gt; 1.Application fraud {@link OriginationErrorConstants.OR_OFFER_FRAUD_WARNING} : FC_OR_OF_0051&lt;br&gt; 2.Party warning indicators for account opening {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040&lt;br&gt; 3.Pending auto refinance policy {@link OriginationErrorConstants#AUTO_REFI_POLICY_PENDING} : FC_OR_OF_0012 &lt;br&gt; 4.No capital subvention details {@link OriginationErrorConstants#CAPITAL_SUBVENTION_VARIANT_DETAILS_NULL} : FC_OR_OF_0011&lt;br&gt; 5.Verification of the application is not yet completed {@link OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} :FC_OR_OF_0047&lt;br&gt; 6.Collateral valuation is incomplete {@link OriginationErrorConstants#OR_OFFER_COL_VALUATION_INCOMPLETE_LIST} :FC_OR_OF_0054&lt;br&gt; 7.Collateral title search is incomplete {@link OriginationErrorConstants#OR_OFFER_TITLE_SEARCH_INCOMPLETE_LIST} :FC_OR_OF_0055&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_APPLN_FOR_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				OFFER} : FC_OR_OF_0041   This error occurs when:  Offer document checklist creation is invoked for a deposit application
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	createDocumentChecklist	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	createDocumentChecklist	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method gets the list of offer document types from DMS for the given checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	createDocumentChecklist	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method gets the list of documents from DMS using the facts in the input document policy detail dto @param sessionContext @param DocumentPolicyDetailDTO [] @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method triggers the creation of offer document checklist for the application. The following steps are performed: 1. If offer documents checklist has already been created for the application, go to step 6 2. Pre create offer validations are performed. 3. If all the validations are successful, the facts required for document generation are collected and passed to DMS 4. Based on the facts, a checklist with a list of documents is created by the DMS module by executing document policies 5. The list of documents along with the checklist ID is returned to origination. 6. The checklist Id is stored in origination for future reference and the list of documents obtained in the DMS response is returned to the UI for generation and dispatch @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @throws FatalException @exception   {@link OriginationErrorConstants#OFFER_GENERATION_POLICY_NA} : FC_OR_OF_0003   This error occurs

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>when:&lt;br&gt; The DMS returns null response after executing policy&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#COLLATERAL_EVALUATION_NOT_DONE} : FC_OR_OF_0010 &lt;br&gt; This error occurs when:&lt;br&gt; Collateral evaluation (valuation and title search) have not been completed for a collateral(s), if any. An overridable policy exception is thrown as an information message is displayed to the customer. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040 &lt;br&gt; This error occurs when any of the validations performed before creating checklist, fail&lt;br&gt; 1.Application fraud {@link OriginationErrorConstants.OR_OFFER_FRAUD_WARNING} : FC_OR_OF_0051&lt;br&gt; 2.Party warning indicators for account opening {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040&lt;br&gt; 3.Pending auto refinance policy {@link OriginationErrorConstants#AUTO_REFI_POLICY_PENDING} : FC_OR_OF_0012 &lt;br&gt; 4.No capital subvention details {@link OriginationErrorConstants#CAPITAL_SUBVENTION_VARIANT_DETAILS_NULL} : FC_OR_OF_0011&lt;br&gt; 5.Verification of the application is not yet completed {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} :FC_OR_OF_0047&lt;br&gt; 6.Collateral valuation is incomplete {@link OriginationErrorConstants#OR_OFFER_COL_VALUATION_INCOMPLETE_LIST} :FC_OR_OF_0054&lt;br&gt; 7.Collateral title search is incomplete {@link OriginationErrorConstants#OR_OFFER_TITLE_SEARCH_INCOMPLETE_LIST} :FC_OR_OF_0055&lt;br&gt;&lt;br&gt; {@link OriginationErrorConstants#INVALID_APPLN_FOR_OFFER} : FC_OR_OF_0041 &lt;br&gt; This error occurs when:&lt;br&gt; Offer document checklist creation is invoked for a deposit application</p>
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This method gets the list of offer document types from DMS for the given checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This service will fetch the document information for the inbound document checklist identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, inboundDocumentChecklistId, , Indicates the inbound document checklist identifier. @return DocumentChecklistResponse, Contains information about documents that are present in the checklist. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocumentChecklist	This method gets the list of document types from DMS for the checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException	This service will populate facts for applicant , product group and collateral (optional) required for Inbound documentation. This information is passed to the Document Management System which evaluates the facts for each applicant for all product groups applied and collateral(s) if submitted for the applicant. It internally creates a set of document identifiers to identify the documents both mandatory and optional to be provided by the applicant. All document identifiers are grouped under a single Document Checklist and the identifier for this checklist is provided to origination. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException @exception   {@link OriginationErrorConstants#FAILED_IN_GEN_INBOUND_CHECKLIST} : FC_OR_DE_0087   This exception is thrown if there is a problem in inbound document checklist generation in DMS.    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateCreditMemorandum	<p>This service is called to generate the credit memorandum document at the Documents module. A list of facts is passed as an input and a documentId is received as response from DMS module. Document type will be set as CreditMemorandum.</p> <p>@fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param  MI,OutboundDocumentRequestDTO,outboundDocumentRequestDTO, ,outboundDocumentRequestDTO  This DTO contains the details required for document generation</p> <p>@return {@link com.ofss.fc.app.origination.dto.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This service generates the credit memorandum identifier @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@return {@link java.lang.String} A string value of generated identifier for the credit decision. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateCreditMemorandum	<p>This service is called to generate the credit memorandum document at the Documents module. A list of facts is passed as an input and a documentId is received as response from DMS module. Document type will be set as CreditMemorandum.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,OutboundDocumentRequestDTO,outboundDocumentRequestDTO, ,outboundDocumentRequestDTO This DTO contains the details required for document generation</p> <p>@return {@link com.ofss.fc.app.origination.dto.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This service is called to save the credit memorandum. This service generates a creditMemorandumId and then updates the input {@link com.ofss.fc.app.origination.dto.lending.core.credit.memorandum.CreditMemorandumDTO} .This service is called after receiving the response from the decision interface and is also called after generating the credit memorandum document. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,CreditMemorandumDTO,creditMemoDTO, ,creditMemoDTO This DTO contains the details related to credit memorandum @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateCreditMemorandum	<p>This service is called to generate the credit memorandum document at the Documents module. A list of facts is passed as an input and a documentId is received as response from DMS module. Document type will be set as CreditMemorandum.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,OutboundDocumentRequestDTO,outboundDocumentRequestDTO, ,outboundDocumentRequestDTO This DTO contains the details required for document generation</p> <p>@return {@link com.ofss.fc.app.origination.dto.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This service is called to generate the credit memorandum document at the Documents module. If defer execution flag is true then request is saved in queue to generate report in async mode. If defer execution is false, this service calls the {@link com.ofss.fc.app.adapter.origination.IOriginationDocumentationAdapter} to generate a document for CreditMemorandum and return a documentId. The document id is persisted in Credit memorandum and history tables.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,Strig,submissionId, , This parameter is the submission id memorandum @fcb.param MI,Boolean,deferExecution, , This indicates if credit memo request needs to be saved in queue or sent to DMS @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	<p>This service will fetch the list of inbound documents grouped under a checklist.&lt;br&gt; Facts are passed for each party applying for a product group. Documents are returned for each party.&lt;br&gt; Similarly, in case of collaterals present, documents are returned for each owner of the collateral&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	@param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException	This service will populate facts for applicant , product group and collateral (optional) required for Inbound documentation. This information is passed to the Document Management System which evaluates the facts for each applicant for all product groups applied and collateral(s) if submitted for the applicant. It internally creates a set of document identifiers to identify the documents both mandatory and optional to be provided by the applicant. All document identifiers are grouped under a single Document Checklist and the identifier for this checklist is provided to origination. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException @exception   {@link OriginationErrorConstants#FAILED_IN_GEN_INBOUND_CHECKLIST} : FC_OR_DE_0087   This exception is thrown if there is a problem in inbound document checklist generation in DMS.    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	<p>This service will fetch the list of inbound documents grouped under a checklist.&lt;br&gt; Facts are passed for each party applying for a product group. Documents are returned for each party.&lt;br&gt; Similarly, in case of collaterals present, documents are returned for each owner of the collateral&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException</p>	<p>This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	getInboundDocumentChecklist	<p>This service will fetch the list of inbound documents grouped under a checklist.&lt;br&gt; Facts are passed for each party applying for a product group. Documents are returned for each party.&lt;br&gt; Similarly, in case of collaterals present, documents are returned for each owner of the collateral&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException</p>	<p>This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This service will update the checklist information. For applicants that are new, applicant serial number is initially used to create inbound document checklist. Once the applicants are on boarded, the party identifiers generated are updated back to the checklist. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistDTO, documentChecklistDTO, , Indicates the document checklist information for update. It contains the party identifiers updated in the checklist documents. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details.   The submission draft can be retrieved in the following ways:   1. My Tracker - This tracker is used by the applicant to log into the system.   2. Application Tracker - This tracker is used by the banking associated to log into the system.   When the submission is saved, all applicants are on-boarded as 'Member' in the party system. The process takes care of on-boarding process  The applicants can later log into the system and submit the application or the banker can retrieve the submission using the applicants and continue with the submission.  The inputs submissionInputDTO will be serialized into String and passed to DraftApplicationService for save as draft.  This service is also called by BPEL process to save the draft submissionInputDTO in the following scenarios-  i) before calling createSubmission   ii) if any error is encountered on confirm of submission.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionProcessDraftDTO, submissionProcessDraftDTO, ,This class holds the entire information of one submission and contains the input details as well as the processed output from the process @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>required information about the status of execution indicating if the invocation succeeded or failed.</p> <p>@exception  OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	<p>This service will create a Submission from the input details .&lt;br/&gt; &lt;br/&gt;The service will be called from the BPEL process. The following activities will be carried out in sequence.&lt;br/&gt; &lt;br/&gt; a) The service creates all new applicants for this submission by iterating through the list of ApplicantData.&lt;br/&gt; &lt;br/&gt; b) This service will create applications based on the Product details submitted by the user.&lt;br&gt; The logic of creating multiple applications is as follows:&lt;br&gt; &lt;br&gt; 1) All deposit product groups applied will be put into separate applications.&lt;br&gt; 2) All lending product groups newly applied will be split into separate applications based on the borrowing entity (group of parties identified by borrowing entity identifier). Different product groups applied by the same borrowing entity will be grouped under the same application.This is applicable for amended facilities as well.&lt;br&gt; 3) All lending product groups that are applied as a result of amending existing facilities will be put into separate applications. &lt;br&gt; &lt;br&gt; c) Simulation details are saved using the corresponding simulation application services based on the product group. &lt;br&gt; For e.g&lt;br&gt; Demand Deposit Product Simulation will be saved using the DemandDepositSimulationApplication Service. &lt;br/&gt; Term Deposit Product Simulation will be saved using</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the TermDepositSimulationApplication Service. &lt;br/&gt; Overdraft Product Simulation will be saved using the DemandDepositSimulationApplication Service. &lt;br/&gt; &lt;br&gt; d) The submission status is marked as SUBMISSION_SUBMITTED.&lt;br&gt; &lt;br&gt; e) While creating a submission, the history records for submission are checked. If a record already exists, the number of history records for that submission identifier are set. This is required as an input to credit decision engine. &lt;br&gt; &lt;br&gt; f) Submission expiry Date is calculated and saved using preference based on Brand and Market Entity.&lt;br&gt; &lt;br&gt; g) For new applicants with generated party identifiers, the new identifiers are replaced in the Inbound Document Checklist.&lt;br&gt; h) In case of bundle applied within the submission, bundle subscription is done for all owners of the submission.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse} This class holds the result of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed and an instance of the SubmissionDTO which contains all information of the created submission. Refer to <a href="#">{@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse}</a> for more details. <a href="#">@throws FatalException @exception</a> <a href="#">{@link OriginationErrorConstants#SUBMISSION_NOT_FOUND}</a> : FC_OR_DE_0055 <a href="#">&lt;br&gt;</a> This error occurs when:<a href="#">&lt;br&gt;</a> The details for the submission identifier were not found or the submission identifier is incorrect.<a href="#">&lt;br&gt;</a> <a href="#">{@link OriginationErrorConstants#OR_INV_OUTPUT}</a> : FC_OR_DE_0064 <a href="#">&lt;br&gt;</a> This error occurs when:<a href="#">&lt;br&gt;</a> The details for applicants that have been created or updated does not match the input list of applicants data<a href="#">&lt;br&gt;</a> <a href="#">{@link OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND}</a><a href="#">&lt;br&gt;</a> FC_OR_DE_0075 error occurs when:<a href="#">&lt;br&gt;</a> The submission history for the submission cannot be fetched.<a href="#">&lt;br&gt;</a> <a href="#">&lt;br&gt;</a></p>
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. <a href="#">@param sessionContext @param documentChecklistDTO @throws FatalException</a>	<a href="#">@param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException</a>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocumentChecklist	This service updates the document checklist with party identifiers when submission is created. @param sessionContext @param documentChecklistDTO @throws FatalException	This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	This method will be called from UI when new verification task will be added in checklist. Internally  it will send an alert to applicant specifying to submit documents for verification if any pending.  The purpose of This method is only for sending an alert if any new verification task is later   added so that we can send alert specifying that documents needs to be verified. @param sessionContext @param verificationDTO @param applicationId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	This service is used to do post verification operation.  For credit card application, this checks for the structure solution confirmation status. If all product groups are confirmed, then invokes the process to process Open Credit Card Account @param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @return TransactionStatus, transactionStatus, , Transaction status of the operation   @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	This service is used to search application / submission summary for the given search criteria input. The method logic is as follows, <ul> <li>Convert ApplicationSearchCriteriaDTO to ApplicationSearchRequestDTO using the builder @link{ApplicationSearchRequestDTOBuilder (invoke the build method)}</li> <li>Search the application by invoking the service, ApplicationSearchApplicationService @link{ApplicationSearchApplicationService#searchApplication}<li>Convert ApplicationSearchResponse to ApplicationSearchSummaryDTO using the builder @link{ApplicationSearchSummaryDTOBuilder by invoking the corresponding build operation.</li> <li>Set the summary DTO in the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p><a href="#">@link{com.ofss.fc.app.workflow.tracker.HumanTaskInquiryResponse}</a>, This contains the information on worklist items / human tasks pending for the application</p> <p><a href="#">@link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType}</a> @throws <a href="#">com.ofss.fc.infra.exception.FatalException</a></p>
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	<p>This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws <a href="#">com.ofss.fc.infra.exception.FatalException</a></p>	<p>This service is used to perform the default search that is a search based on applicationId, submission Id or partyId &lt;ul&gt;</p> <p>&lt;li&gt;<a href="#">@link{com.ofss.fc.app.Origination.dto.core.search.input.ApplicationSearchRequestDTO}</a> is built using the builder</p> <p><a href="#">@link{com.ofss.fc.app.Origination.service.core.dto.builder.application.tracking.ApplicationSearchRequestDTOBuilder}</a>&lt;/li&gt;</p> <p>&lt;li&gt;<a href="#">ApplicationSearchRequestDTO</a> used to fetch <a href="#">@link{com.ofss.fc.app.Origination.dto.core.search.output.BasicApplicationSearchResponse}</a> through the service</p> <p><a href="#">@link{com.ofss.fc.app.Origination.service.core.search.ApplicationSearchApplicationService.performDefaultSearch(SessionContext, ApplicationSearchRequestDTO ) }</a></p> <p>&lt; / li &gt; &lt;li&gt;</p> <p><a href="#">@link{com.ofss.fc.app.Origination.dto.core.search.application.tracking.ApplicationSearchSummaryDTO }</a> is created using builder <a href="#">@link{com.ofss.fc.app.Origination.service.core.dto</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre>         builder application         tracking         BasicApplicationSearchSummaryDTOBuilder         BasicApplicationSearchSummaryDTOBuilder         ( ) } &lt;/li&gt; &lt;li&gt;         @link{ com ofss         fc app origination         dto core         application tracking         ApplicationSearchSummaryResponse }         is set and returned         &lt;/li&gt; &lt; / ul &gt;         @fcb.param MI,SessionContext,sessionContext,Session         Context which must contain the session context         information such as user id, branch, branch code,         channel etc @fcb.param         MI,@link{com.ofss.fc.app.origination.dto.core.applicati         on.tracking.ApplicationSearchCriteriaDTO}, contains         the criterion on which applications are to be searched         @return         @link{com.ofss.fc.app.origination.dto.core.application.         tracking.ApplicationSearchSummaryResponse}         @throws com.ofss.fc.infra.exception.FatalException </pre>
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	This method searches for all the documents having the primary auxiliary index as the submissionId given in input @param submissionId @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	search	This method searches for a document using the specified fields with an or condition. @param sessionContext @param documentSearchDTO @return @throws com.ofss.fc.infra.exception.FatalException	This method searches for all the documents having the primary auxiliary index as the submissionId given in input @param submissionId @param sessionContext @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateDocument	<p>This service is called to generate the application form document at the Documents module. A list of facts is passed as an input and a documentId is received as response from the DMS module. @param sessionContext @param outboundDocumentRequestDTO @return DocumentGenerationResponse @throws FatalException</p>	<p>This method is used to generate the application form based on the product group serial number and the submissionInputDTO which includes the submission details such as submission ID, ApplicantDataDTO and ProductDetailsDTO.&lt;br&gt; &lt;br&gt; A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the document is to be generated.In case it is null, the document will be generated for all the product group serial numbers of the given submissionInputDTO. @return {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} This class holds an array of the ApplicationFormDocumentDTO which contains the documentId of the newly created document with its submissionId and product group serial number. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} for more details. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateDocument	This service is called to generate the application form document at the Documents module. A list of facts is passed as an input and a documentId is received as response from the DMS module. @param sessionContext @param outboundDocumentRequestDTO @return DocumentGenerationResponse @throws FatalException	This service is called to generate the credit memorandum document at the Documents module. A list of facts is passed as an input and a documentId is received as response from DMS module. Document type will be set as CreditMemorandum. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OutboundDocumentRequestDTO,outboundDocumentRequestDTO, ,outboundDocumentRequestDTO This DTO contains the details required for document generation @return {@link com.ofss.fc.app.origination.dto.core.credit.memorandum.CreditMemorandumGenerationResponse} This object contains the response of this service that includes creditMemorandumId and documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	generateDocument	This service is called to generate the application form document at the Documents module. A list of facts is passed as an input and a documentId is received as response from the DMS module. @param sessionContext @param outboundDocumentRequestDTO @return DocumentGenerationResponse @throws FatalException	This service is called to generate the application form document at the Documents module. A list of facts is passed as an input and a documentId is received as response from the DMS module. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OutboundDocumentRequestDTO,outboundDocumentRequestDTO, ,outboundDocumentRequestDTO This DTO contains the details required for document generation @return DocumentGenerationResponse This object contains the response of this service that includes the documentId of generated document. @throws com.ofss.fc.infra.exception.FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method is used to generate the application form based on the product group serial number and the submissionInputDTO which includes the submission details such as submission ID, ApplicantDataDTO and ProductDetailsDTO.    A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document.     @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, Array contains all the product group serial numbers for which the document is to be generated.In case it is null, the document will be generated for all the product group serial numbers of the given submissionInputDTO. @return {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} This class holds an array of the ApplicationFormDocumentDTO which contains the documentId of the newly created document with its submissionId and product group serial number. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} for more details. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method is used to generate the secondary auxiliary indexes for an already generated document based on the product group serial number and the submissionInputDTO which includes the ProductDetailsDTO from which we obtain the document details for the various products.    A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document.     @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the indexes are to be generated. @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method updates the AuxillaryDocumentIndexDTO for all the documents within a submission for which the secondary auxiliary indexes are generated by fetching such documents through their document Id's. @param sessionContext @param indexDTOMap @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method triggers the creation of offer document checklist for the application. The following steps are performed: 1. If offer documents checklist has already been created for the application, go to step 6 2. Pre create offer validations are performed. 3. If all the validations are successful, the facts required for document generation are collected and passed to DMS 4. Based on the facts, a checklist with a list of documents is created by the DMS module by executing document policies 5. The list of documents along with the checklist ID is returned to origination. 6. The checklist Id is stored in origination for future reference and the list of documents obtained in the DMS response is returned to the UI for generation and dispatch @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @throws FatalException @exception   {@link OriginationErrorConstants#OFFER_GENERATION_POLICY_NA} : FC_OR_OF_0003   This error occurs when:  The DMS returns null response after executing policy    {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057   This error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.    {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084   This error occurs when:  Warning indicators which prevent account opening are present for applicants applying to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#COLLATERAL_EVALUATION_NOT_DONE} : FC_OR_OF_0010 &lt;br&gt; This error occurs when:&lt;br&gt; Collateral evaluation (valuation and title search) have not been completed for a collateral(s), if any. An overridable policy exception is thrown as an information message is displayed to the customer. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040 &lt;br&gt; This error occurs when any of the validations performed before creating checklist, fail&lt;br&gt; 1.Application fraud {@link OriginationErrorConstants.OR_OFFER_FRAUD_WARNING} : FC_OR_OF_0051&lt;br&gt; 2.Party warning indicators for account opening {@link OriginationErrorConstants#PREOFFER_VALIDATIONS} : FC_OR_OF_0040&lt;br&gt; 3.Pending auto refinance policy {@link OriginationErrorConstants#AUTO_REFI_POLICY_PENDING} : FC_OR_OF_0012 &lt;br&gt; 4.No capital subvention details {@link OriginationErrorConstants#CAPITAL_SUBVENTION_VARIANT_DETAILS_NULL} : FC_OR_OF_0011&lt;br&gt; 5.Verification of the application is not yet completed {@link OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} :FC_OR_OF_0047&lt;br&gt; 6.Collateral valuation is incomplete {@link OriginationErrorConstants#OR_OFFER_COL_VALUATION_INCOMPLETE_LIST} :FC_OR_OF_0054&lt;br&gt; 7.Collateral title search is incomplete {@link OriginationErrorConstants#OR_OFFER_TITLE_SEARCH_INCOMPLETE_LIST} :FC_OR_OF_0055&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_APPLN_FOR_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				OFFER} : FC_OR_OF_0041   This error occurs when:  Offer document checklist creation is invoked for a deposit application
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method updates the execution Status of an offer document. The document is generated and stored in DMS and the corresponding ID is stored in Origination module. During the Acceptance and Execution of Documents, the status of the offer document will be updated. updateOfferDocumentStatus method will call the corresponding service in DMS to update the execution status of the document. @param sessionContext @param OutboundDocumentListRecordDTO @param status @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method gets the list of offer document types from DMS for the given checklist ID @param sessionContext @param outboundDocumentListId - checklist ID @param OutboundDocumentListInquiryResponse response - contains a list of document types @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service will fetch the document information for the inbound document checklist identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, inboundDocumentChecklistId, , Indicates the inbound document checklist identifier. @return DocumentChecklistResponse, Contains information about documents that are present in the checklist. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service will fetch the list of inbound documents grouped under a checklist.  Facts are passed for each party applying for a product group. Documents are returned for each party.  Similarly, in case of collaterals present, documents are returned for each owner of the collateral  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistInputDTO, documentChecklistInputDTO, , Indicates the DTO containing facts for each party for all product groups applied. @return DocumentChecklistResponse, Returns a unique identifier for inbound checklist containing documents to be submitted. @throws com.ofss.fc.infra.exception.FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, , This DTO represents the document. @return DocumentInquiryResponse This data transfer object contains the DocumentDTO which is used to hold the information of document such as document id, document policy id and document type id. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service is called to fetch the document type from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This method updates all affiliates' documents - owner id and document indexes - index values with the appropriate party id. Originally, for indexes - index values, the party id recorded is submissionId + underscore + partySerialNumber. @param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service will populate facts for applicant , product group and collateral (optional) required for Inbound documentation. This information is passed to the Document Management System which evaluates the facts for each applicant for all product groups applied and collateral(s) if submitted for the applicant. It internally creates a set of document identifiers to identify the documents both mandatory and optional to be provided by the applicant. All document identifiers are grouped under a single Document Checklist and the identifier for this checklist is provided to origination. @param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param MI,SubmissionInputDTO, submissionInputDTO, This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException @exception   {@link OriginationErrorConstants#FAILED_IN_GEN_INBOUND_CHECKLIST} : FC_OR_DE_0087   This exception is thrown if there is a problem in inbound document checklist generation in DMS.    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	fetchDocument	This service is called to fetch the document from the Documents module. DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return DocumentInquiryResponse @throws FatalException	This service will inquire the submission with the given application Id and update the status of submission to MARKED_FOR_EDIT.  This service will be called from the process to update the status of submission when user chooses to edit the submission.  Edit submission will be allowed only if any of the following conditions are satisfied-   1) Lending application under the submission has the following status-  <ul> <li>APPLICATION_SUBMITTED</li> <li>APPLICATION_APPROVED</li> <li>APPLICATION_DOWNSELL_REJECTED</li> <li>APPLICATION_MANUAL_DECISION_REJECTED</li> </ul> 2) Lending application under the submission has the following status and posting date is less than reject offer expiry date -  <ul> <li>APPLICATION_REJECT_OFFER_DISPATCHED</li> <li>DECLINE_LETTER_DISPATCHED</li> </ul> 3)Credit card application under the submission has the following status-  <ul> <li>APPLICATION_SUBMITTED</li> <li>APPLICATION_APPROVED</li> <li>APPLICATION_MANUAL_DECISION_REJECTED</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,submissionId, The id of the submission to be inquired @return An instance of {@link TransactionStatus} containing information about the submission. @throws FatalException @exception   {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055   This error occurs when:  The details for the submission identifier were not found or the submission identifier is

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				incorrect.    {@link OriginationErrorConstants#SUBMISSION_MARKED_FOR_EDIT_WITHDRAW} : FC_OR_DE_0090   This error occurs when:  The submission trying to be edited has been already marked for edit    {@link OriginationErrorConstants#SUBMISSION_CANNOT_BE_EDITED} : FC_OR_DE_0091   This error occurs when:  The submission being marked for edit has no product groups that can be edited.  This will happen if all accounts under the product group are fulfilled, rejected or withdrawn.    {@link OriginationErrorConstants#SUBMISSION_CANNOT_BE_EDITED_OFFER_EXPIRED} : FC_OR_DE_0090   This error occurs when the posting date is after the reject offer expiry date. 
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method updates the AuxillaryDocumentIndexDTO for all the documents within a submission for which the secondary auxiliary indexes are generated by fetching such documents through their document Id's. @param sessionContext @param indexDTOMap @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method updates the execution Status of an offer document. The document is generated and stored in DMS and the corresponding ID is stored in Origination module. During the Acceptance and Execution of Documents, the status of the offer document will be updated. updateOfferDocumentStatus method will call the coressponding service in DMS to update the execution status of the document. @param sessionContext @param OutboundDocumentListRecordDTO @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				status @throws FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This service will update the checklist information. For applicants that are new, applicant serial number is initially used to create inbound document checklist. Once the applicants are on boarded, the party identifiers generated are updated back to the checklist. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DocumentChecklistDTO, documentChecklistDTO, , Indicates the document checklist information for update. It contains the party identifiers updated in the checklist documents. @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This service is called to update the document at the Documents module. DocumentDTO is passed as an input. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, , This DTO represents the document. @return DocumentContentResponse This data transfer object is use to hold the information of document such as document id, content reference id, and archival status flag. @throws com.ofss.fc.infra.exception.FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details.   The submission draft can be retrieved in the following ways:   1. My Tracker - This tracker is used by the applicant to log into the system.   2. Application Tracker - This tracker is used by the banking associated to log into the system.   When the submission is saved, all applicants are on-boarded as 'Member' in the party system. The process takes care of on-boarding process  The applicants can later log into the system and submit the application or the banker can retrieve the submission using the applicants and continue with the submission.  The inputs submissionInputDTO will be serialized into String and passed to DraftApplicationService for save as draft.  This service is also called by BPEL process to save the draft

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>submissionInputDTO in the following scenarios-&lt;br&gt;  i) before calling createSubmission &lt;br&gt; ii) if any error is encountered on confirm of submission.&lt;br&gt;  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionProcessDraftDTO, submissionProcessDraftDTO, ,This class holds the entire information of one submission and contains the input details as well as the processed output from the process @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This service will create a Submission from the input details .   The service will be called from the BPEL process. The following activities will be carried out in sequence.    a) The service creates all new applicants for this submission by iterating through the list of ApplicantData.    b) This service will create applications based on the Product details submitted by the user.  The logic of creating multiple applications is as follows:    1) All deposit product groups applied will be put into separate applications.  2) All lending product groups newly applied will be split into separate applications based on the borrowing entity (group of parties identified by borrowing entity identifier). Different product groups applied by the same borrowing entity will be grouped under the same application.This is applicable for amended facilities as well.  3) All lending product groups that are applied as a result of amending existing facilities will be put into separate applications.     c) Simulation details are saved using the corresponding simulation application services based on the product group.   For e.g  Demand Deposit Product Simulation will be saved using the DemandDepositSimulationApplication Service.   Term Deposit Product Simulation will be saved using the TermDepositSimulationApplication Service.   Overdraft Product Simulation will be saved using the DemandDepositSimulationApplication Service.     d) The submission status is marked as SUBMISSION_SUBMITTED.    e) While creating a submission, the history records for submission are checked. If a record already exists, the number of history records for that submission identifier are set. This is required as an input to credit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>decision engine. &lt;br&gt; &lt;br&gt; f) Submission expiry Date is calculated and saved using preference based on Brand and Market Entity.&lt;br&gt; &lt;br&gt; g) For new applicants with generated party identifiers, the new identifiers are replaced in the Inbound Document Checklist.&lt;br&gt; h) In case of bundle applied within the submission, bundle subscription is done for all owners of the submission.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse} This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed and an instance of the SubmissionDTO which contains all information of the created submission. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse} for more details. @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> <a href="#">FatalException</a> @exception &lt;br&gt; {<a href="#">@link</a> <a href="#">OriginationErrorConstants#SUBMISSION_NOT_FOUND</a>} : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {<a href="#">@link</a> <a href="#">OriginationErrorConstants#OR_INV_OUTPUT</a>} : FC_OR_DE_0064 &lt;br&gt; This error occurs when:&lt;br&gt; The details for applicants that have been created or updated does not match the input list of applicants data&lt;br&gt; &lt;br&gt; {<a href="#">@link</a> <a href="#">OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND</a>}&lt;br&gt; FC_OR_DE_0075 error occurs when:&lt;br&gt; The submission history for the submission cannot be fetched.&lt;br&gt; &lt;br&gt; </p>
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method updates all affiliates' documents - owner id and document indexes - index values with the appropriate party id. Originally, for indexes - index values, the party id recorded is submissionId + underscore + partySerialNumber. @param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	@param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	updateDocument	This service is called to update the document at the Documents module. DocumentDTO is passed as an input . @param sessionContext @param documentDTO @return DocumentContentResponse @throws FatalException	This method is used to update document list for new Associated Party @param sessionContext @param accountId @param submissionId @param checklistId @throws FatalException
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	validateMandatoryDocuments		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	deleteDocument	This service is called to delete the document(s). DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return TransactionStatus @throws FatalException	This service is called to delete the document(s). DocumentDTO is passed as an input. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, , This DTO represents the document. @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
IOrigination DocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	deleteDocument	This service is called to delete the document(s). DocumentDTO is passed as an input. @param sessionContext @param documentDTO @return TransactionStatus @throws FatalException	This service will inquire the submission with the given application Id and update the status of submission to MARKED_FOR_EDIT.  This service will be called from the process to update the status of submission when user chooses to edit the submission.  Edit submission will be allowed only if any of the following conditions are satisfied-   1) Lending application under the submission has the following status-  <ul> <li>APPLICATION_SUBMITTED</li> <li>APPLICATION_APPROVED</li> <li>APPLICATION_DOWNSELL_REJECTED</li> <li>APPLICATION_MANUAL_DECISION_REJECTED</li> </ul> 2) Lending application under the submission has the following status and posting date is less than reject offer expiry date -  <ul> <li>APPLICATION_REJECT_OFFER_DISPATCHED</li> <li>DECLINE_LETTER_DISPATCHED</li> </ul> 3)Credit card application under the submission has the following status-  <ul> <li>APPLICATION_SUBMITTED</li> <li>APPLICATION_APPROVED</li> <li>APPLICATION_MANUAL_DECISION_REJECTED</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>such as user id, branch, branch code, channel etc.  @fcb.param MI,String,submissionId, ,The id of the submission to be inquired @return An instance of {@link TransactionStatus} containing information about the submission. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_MARKED_FOR_EDIT_WITHDRAW} : FC_OR_DE_0090 &lt;br&gt; This error occurs when:&lt;br&gt; The submission trying to be edited has been already marked for edit&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_CANNOT_BE_EDITED} : FC_OR_DE_0091 &lt;br&gt; This error occurs when:&lt;br&gt; The submission being marked for edit has no product groups that can be edited.&lt;br&gt; This will happen if all accounts under the product group are fulfilled, rejected or withdrawn.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_CANNOT_BE_EDITED_OFFER_EXPIRED} : FC_OR_DE_0090 &lt;br&gt; This error occurs when the posting date is after the reject offer expiry date.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	searchForIndex	<p>This method searches for a document using the specified fields with an or condition. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,documentDTO TODO: Explain the documentDTO above @fcb.param MI,List&lt;String&gt;,contentIdList, ,A list of content ids to be used for searching. @return {@link com.ofss.fc.app.content.capture.dto.DocumentSearchResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDocumentationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDocumentationAdapter	searchForIndex	<p>This method searches for a document using the specified fields with an or condition. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,documentDTO TODO: Explain the documentDTO above @fcb.param MI,List&lt;String&gt;,contentIdList, ,A list of content ids to be used for searching. @return {@link com.ofss.fc.app.content.capture.dto.DocumentSearchResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	<p>This method will return a hash map containing all the parties ' id in the application and the corresponding party names. @param applicationId @return HashMap @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	<p>@param sessionContext @param accountId @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This is called from inquireSubmissionDecision and used to populate amounts in party debt servicing and debt servicing group details @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,DebtServicingGroupDTO,debtServicingGroupDTOs @fcb.param MI,SubmissionDecision,submissionDecision @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	@param sessionContext @param applicationId @param borrowingEntity @param applicantDecisionDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantProductGroupReferenceDTO[] applicantsForPG, , Indicates the applicants of the product group @return List<ValidationError>, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String[] partyIds, , Indicates the party id linked to account @return List<ValidationError>, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OrigationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;</p>	<p>Fetch party names for the given party ids @param partyIds @return partyNames</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt;{@link OrigationErrorConstants#INVALID_PARTY_ID}&lt;br&gt;FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;</p>	<p>This method will be called from UI to fetch Basic CCI Details for Applicant to be shown in CCI Human task as grid. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;</p>	<p>This will fetch all CCI details available under given application. @param sessionContext, contains user information. @param applicationId, denotes the application ID @throws FatalException {@link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This service will add error for each warning indicator. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, partyId, , Indicates the party for which applicant warnings are being checked. @fcb.param MI,String, eventCode, , Indicates the event for which warning indicators are being checked. @fcb.param MI,ModuleType, moduleType, , Indicates the event is being fetched for which Product group. @fcb.param MI,String, productGroupId, , Indicates the product group identifier @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OrigationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;</p>	<p>This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	<p>This service will fetch the party's name from party module.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID}&lt;br&gt; FC_OR_DE_0057 error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt;&lt;br&gt;&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicantName	This service will fetch the party's name from party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, applicantId, , Indicates the identifier for the party @return String, Indicates the name for the party @throws FatalException  {@link OriginationErrorConstants#INVALID_PARTY_ID} FC_OR_DE_0057 error occurs when:  The details for the party identifier cannot be fetched or the party identifier is incorrect.  	This method does a validation for the PartyID. It checks whether the partyId is equal to null or empty. Then it queries the PartyModule to get the name for that party ID. If the name exists in the Party Module, and the name is not empty, the partyId is for sure a valid ID. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, The input DTO of the submission to be validated. @return boolean, boolean value to indicate whether the partyId is valid. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	The method originateDebitCard is responsible for actually doing debit card request   The actual implementation of debit card will be based on runtime configuration and will be done via adapter methodology   @param sessionContext @param debitCardRequestDTO @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This method is used to build ProductGroupApprovalDataDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service is used to fetch the eligible offers for the selected product group from PMU module.  An eligibility check is run based on application channel, applicants applying for the product group and facility parameters.  A list of offers complying with the above conditions are returned. @param sessionContext @param partyId @param productGroupDTO @param submissionHeaderDTO @return OfferInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service will fetch the guarantors for a particular facility. It will check with LCM for the guarantee collaterals linked to this facility and then for all those owners fetch the party details from MDM @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return MDMPartiesInquiryResponse , Response containing details of all guarantors for the facility. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyDetails	This service will fetch the detailed information from party module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This method is to validate the party information for an ApplicantDTO object. For a new applicant who is an EXISTING_PARTY, we need to validate the party_id. And, if the other information is not input, we need to populate them. This method input ApplicantDataDTO @param sessionContext @param applicantDataDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service will populate the Applicant Data DTO using the party identifier of the applicant.  The party details are fetched using the MDM service. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDTO applicantDTO, , The applicant information for which details are being populated. @return ApplicantDataDTO, Returns applicant information. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service will populate the Applicant Data DTO from Applicant DTO @param sessionContext @param applicantDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This method is used to fetch a list of applicants by passing an applicationId. NOTE: Caller must make sure that the input applicationId is validated. The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception     {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052   This error occurs when:  The application identifier is invalid and no application details are found for the same. 



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	<p>This service will fetch the detailed information from MDM module for the party identifier passed.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException</p>	<p>This method is used to fetch a list of applicants and affiliates for the input applicationId. &lt;br&gt; NOTE: Caller must make sure that the input applicationId is validated. &lt;br&gt; The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. The related affiliates are also added to the list of applicants. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052 &lt;br&gt; This error occurs when:&lt;br&gt; The application identifier is invalid and no application details are found for the same.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service fetches party details and populates factsMap to execute condition/covenant policy @param sessionContext @param applicantDecisionDTO @param borrowingEntityId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	The method will construct facts for the primary applicant in the application @fcb.param MI, applicationId - Application details which will be used to construct the factsDTO. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	Construct facts for part from Origination fact application (MDM party facts) @param partyFacts @param applicantId @param partyId @param sessionContext @return String @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchMDM PartyDetails	This service will fetch the detailed information from MDM module for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed response for the party. @throws FatalException	This service will fetch the guarantors for a particular facility. It will check with LCM for the guarantee collaterals linked to this facility and then for all those owners fetch the party details from MDM @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return MDMPartiesInquiryResponse , Response containing details of all guarantors for the facility. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isCustomer	This service will validate if the party has the role of Customer in party module. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return boolean, Indicates if the party is a customer or not. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This method constructs the TD account opening DTO @param sessionContext @param accountConfigDTO @param applicationId @return TermDepositAccountOpenDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service will create a simulation or save an existing simulation for overdraft products being setup for an application. Each overdraft product group is attached to a facility maintained in LCM system. In case of overdraft product groups, the user can setup multiple simulations for different offers selected. Each offer corresponds to a single simulation and contains the account configuration and additional details. The primary applicant will provide various preference details such as cheque book preference, deposit book preference, etc for the account. Preferences are also captured for each applicant of the account for statement, debit card details, etc. Overdraft facility can also be opted for existing demand deposit accounts in which case the user does not have to provide the configuration details since the account is already active. An existing demand deposit account can be attached only if the following conditions are fulfilled:     Parties applying for Overdraft

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners.</p> <p>&lt;br&gt; &lt;b&gt;For e.g.&lt;/b&gt; : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list. &lt;br&gt; Existing CASA account selected for Overdraft, must not be linked to any other Overdraft facility. &lt;br&gt; &lt;br&gt; The system will internally create a linkage for this simulation to be fetched for future retrieval. The applicant(s) can continue making changes to this simulation until the account is opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String submissionId, , Indicates the identifier of the submission for which the the simulation details are being saved @fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application. @fcb.param MI,ODSimulationDTO, odSimulationDTO, , This class holds simulation details for the overdraft product. It contains information about the accounts being configured within the simulation such as statement preferences, card preferences, etc. @return {@link ODSimulationResponse} This class contains the simulation details that passed and further updated within this service. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service find out MinorAccount status of an account using the date of birth of the primary applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and flgoperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception   {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074   This error occurs when:  The primary applicant information for which minor status is being fetched is not found.    {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089   This error occurs when:  The birthdate for the primary applicant cannot be retrieved.   



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service find out MinorAccount status of an account using the date of birth of the applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and flgoperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception   {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074   This error occurs when:  The primary applicant information for which minor status is being fetched is not found.    {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089   This error occurs when:  The birthdate for the primary applicant cannot be retrieved.   

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service updates the party Id in product group linkages. @param linkageDTO @param applicantMap @param applicantSerialNumber @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service populates the party information in DD account configurations. @param ddaConfiguration @param applicantProductGroupReferenceDTOs @param applicantMap @param primaryApplicantPartyId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatus	This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @fcb.param MI, ProductClass, productClass, , Indicates the productClass @return AgeStatus,Indicates the minor/major status for a party @throws FatalException	This service will create default account holder preferences for Corporate and Trust Parties.  This is required for adding party to account relationships for corporate and trust parties.  @fcb.param MI,Set<IApplicantProductGroupReference> applicantProductGroupReferences, , Indicates the applicants of the product group. @fcb.param MI,DepositAccountConfigurationDTO depositAccountConfigurationDTO, , Indicates the deposit account configuration details. @fcb.param MI,String offerName, , Indicates the offer name @return List<DepositAccountHolderPreferenceDTO>, Returns the updates deposit account holder preferences for the account configuration. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatusBasedonAge	<p>This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, PartyDataDTO @return PartyAccountStatusDTO,Indicates the details minor/major status for a party with respect to an account @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} &lt;br&gt; FC_OR_DE_0074 error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt;</p>	<p>This service find out MinorAccount status of an account using the date of birth of the primary applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and flgoperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074 &lt;br&gt; This error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089 &lt;br&gt; This error occurs when:&lt;br&gt; The birthdate for the primary applicant cannot be retrieved.&lt;br&gt; &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyMinorStatusBasedonAge	<p>This service will return the age status for the party identifier passed. It indicates if the party is a minor or not. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, PartyDataDTO @return PartyAccountStatusDTO,Indicates the details minor/major status for a party with respect to an account @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} &lt;br&gt; FC_OR_DE_0074 error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt;</p>	<p>This service find out MinorAccount status of an account using the date of birth of the applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and flgoperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074 &lt;br&gt; This error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089 &lt;br&gt; This error occurs when:&lt;br&gt; The birthdate for the primary applicant cannot be retrieved.&lt;br&gt; &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchIndividualDetails	This service will fetch the individual details for a party of the type individual. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed party response for an individual party. @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchIndividualDetails	This service will fetch the individual details for a party of the type individual. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return PartyInquiryResponse, Indicates the detailed party response for an individual party. @throws FatalException	This method is to validate the party information for an ApplicantDTO object. For a new applicant who is an EXISTING_PARTY, we need to validate the party_id. And, if the other information is not input, we need to populate them. This method input ApplicantDataDTO @param sessionContext @param applicantDataDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt;Else return false indication KYC is not done.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @throws FatalException</p>	<p>This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantProductGroupReferenceDTO[] applicantsForPG, , Indicates the applicants of the product group @return List&lt;ValidationError&gt;, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt;Else return false indication KYC is not done.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @throws FatalException</p>	<p>This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String[] partyIds, , Indicates the party id linked to account @return List&lt;ValidationError&gt;, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;li&gt;Else return false indication KYC is not done.</p> <p>@fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @throws FatalException</p>	<p>This alerts will be send to all applicants during account opening for CASA and TD applications.&lt;br&gt; Alert will be send to all parties linked to account.&lt;br&gt; While sending alerts to condition will be checked.</p> <p>&lt;ol&gt; &lt;li&gt;If PartyAccountRelationshipType is not PartyAccountRelationshipType.OTHERS send alert will be send to that party.&lt;/li&gt; &lt;li&gt;If PartyAccountRelationshipType is PartyAccountRelationshipType.OTHERS then alert will be send only if for given affiliate alert to be send is allowed or not.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param originationActivityDTO @param applicationContext</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;li&gt;Else return false indication KYC is not done.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @throws FatalException</p>	<p>Check if KYC is not done for any party @param sessionContext @param applicantDTOs @return boolean @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt;Else return false indication KYC is not done.</p> <p>@fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI, String, partyId, , Indicates the identifier for the party @throws FatalException</p>	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt;Else return false indication KYC is not done. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllAccountsOfParty	This service will fetch all the accounts for the party identifier passed. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the party @return List<String>, Indicates a list of accounts identifiers that are related to the party. @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	createNewMember	If the application is initiated by a new customer whos is not a member already, then the customer needs to be on boarded as a member This service is used to on board the new customer as a member @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, MemberOnboardingDTO, memberDTO, , Indicates the member on boarding details DTO @return {@link MemberOnboardingResponse}, Indicates the status of the on boarding the operation	If the application is initiated by a new customer whos is not a member already, then the customer needs to be on boarded as a member This service is used to on board the new customer as a member @param sessionContext @param memberDTO @return {@link MemberOnboardingResponse}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForApplicant	This service will fetch the party 's current financial profile @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String partyId , Indicates the party @return @throws FatalException	This service will populate the Applicant Data DTO using the party identifier of the applicant.  The party details are fetched using the MDM service. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDTO applicantDTO, , The applicant information for which details are being populated. @return ApplicantDataDTO, Returns applicant information. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForApplicant	This service will fetch the party 's current financial profile @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String partyId , Indicates the party @return @throws FatalException	This service will populate the Applicant Data DTO from Applicant DTO @param sessionContext @param applicantDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchWarningIndicatorForAnEvent	This service will fetch the warning indicators for a party for the event passed. For Origination, event to prevent account opening is passed. @param sessionContext @param partyId @param eventCode @return @throws FatalException	This service will add error for each warning indicator. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, partyId, , Indicates the party for which applicant warnings are being checked. @fcb.param MI,String, eventCode, , Indicates the event for which warning indicators are being checked. @fcb.param MI,ModuleType, moduleType, , Indicates the event is being fetched for which Product group. @fcb.param MI,String, productGroupId, , Indicates the product group identifier @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationshipsForExistingParty		This method is called from the service fetchRelationshipsForExistingParty. It fetches the related party details from party direct relationship table. @param SessionContext @param partyIdList
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationshipsForExistingParty		This method is used to fetch the related parties for the given parties and also the relationship between the parties. This method is called the applicant details screen. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String[],partyIdList, , List of party id's. The identifier which is used to uniquely identify a party with in the system. @return {@link FetchRelatedPartyInquiryResponse} An instance of the Fetch Related Party Inquiry Response which contains the relationship information that exist between two parties @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationShipBetweenParties		This service will populate the fact required for the Employee rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant for the product group @fcb.param MI,ApplicantRelationshipDTO[], applicantRelationshipDTO, , Indicates the information about relationship between applicants of a submission. @fcb.param MI,Map<Integer, ApplicantDataDTO>applicantMap, , Indicates map of applicant data with serial number as the key. @return FactInformationDTO[], List of facts to be passed for Employee Rule Execution @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationShipBetweenParties		This method will fetch the party to party relationships between all parties for the given borrowing entity id @param sessionContext the {@link SessionContext} @param borrowingEntityId the borrowing entity id @return an array of {@link PartyToPartyRelationshipDTO} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationShipBetweenParties		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartiesOfCASA	This service returns the list of parties that are account holders of the account for which the identifier is passed. @param sessionContext @param accountId @return @throws FatalException	This service returns the list of parties that are account holders of the account for which the identifier is passed. @param sessionContext @param accountId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method will create InsuranceDTO from CCIDetailDTO @param cciDetail @param lendingPG @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method will create InsuranceDTO from CCIDetailDTO @param cciDetail @param lendingPG @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	@param sessionContext @param applicationContext @param originationActivityDTO @param accountHolder @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This is continuation method for sending alerts to applicants for CASA and TD Alert. Based on PartyType the specific event will be set. @param sessionContext @param applicationContext @param originationActivityDTO @param partyId @param partyType
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method is used to send alerts to all parties linked with specific party as owner , administrator or others etc. @param applicationContext @param originationActivityDTO @param sessionContext @param partyId



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This is alert send for Lending Account Creation. For account, it will take corresponding simulationID and fetch the LoanAccountConfigurationDTO. <ol><li>If PartyAccountRelationshipType is not PartyAccountRelationshipType.OTHERS send alert will be send to that party.</li><li>If PartyAccountRelationshipType is PartyAccountRelationshipType.OTHERS then alert will be send only if for given affiliate alert to be send is allowed or not.</li></ol> @param sessionContext @param accountOpeningDetailsDTO @param prodGroupSerialNumber @param applicationId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method is to validate the party information for an ApplicantDTO object. For a new applicant who is an EXISTING_PARTY, we need to validate the party_id. And, if the other information is not input, we need to populate them. @param sessionContext @param applicantDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This service will called before 'createCustomer', it will fetch the applicant-data-dto for a given SubmissionId and applicantSerialNumber. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param @fcb.param MI,String,submissionId, ,The submission-id for which the applicant information is to be updated @fcb.param @fcb.param MI,Integer,applicantSerialNumber, ,The applicant serial number for which the applicant information is to be fetched @return An instance of {@link com.ofss.fc.app.origination.dto.core.application.ApplicantInquiryResponse} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	@param sessionContext @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId Input to This method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method will link the applicants to the product group to which they were linked at the time of submission. @param sessionContext the {@link SessionContext} @param accountId the Account Id @param applicantDataList the List of Applicants @param productGroupLinkageDTO the Product Group Linkage @return the {@link ProductGroupLinkageDTO} with applicants linked @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId Input to This method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility. An existing demand deposit account can be used only if the following conditions are fulfilled: Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners. For e.g. : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list. Also, existing CASA selected, must not be linked to any other Overdraft facility. @param MI, SessionContext, sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception {@link OriginationErrorConstants#OR_OFFER_INVALID} :

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FC_OR_DE_0081   This error occurs when:  The offer details are being fetched for incorrect offer identifier or offer has expired.   
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	validateAndfetchAgentDetails		@param submissionHeaderDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method would return {@link ApplicationDecisionDTO}. During the creation of the DTO the factory (instance of {@link ICreditDecisionFactory} would be invoked for values to be set in the ApplicationDecisionDTO object @return ApplicantDecisionDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		<p>This method is used to fetch a list of applicants and affiliates for the input applicationId. &lt;br&gt; NOTE: Caller must make sure that the input applicationId is validated. &lt;br&gt; The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. The related affiliates are also added to the list of applicants. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052 &lt;br&gt; This error occurs when:&lt;br&gt; The application identifier is invalid and no application details are found for the same.&lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method will populate the isShareIncomeFlag and isShareExposureFlag of the applicant decision dto <1> Main applicant, both flag populated as true <2> Direct affiliate, populated based on the flag value in the party to party relationship <3> Indirect affiliate, populated as true only when the flag value of itself and its upper level affiliate are true @param sessionContext @param submissionDTO @param submissionDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This function is used to send alerts for KYC Pending in case of Trust and Corporate case. In this case individual affiliates linked to this organisation/trust and linked to account while creation will get alert. @param sessionContext @param originationActivityDTO @param accountHolders @param partyId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method is used to send alerts to all parties linked with specific party as owner , administrator or others etc. @param applicationContext @param originationActivityDTO @param sessionContext @param partyId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method is used to fetch relationship DTO for related parties with class, Owner / Administrator and relation type, Business. @param sessionContext @param partyId @return List<RelationshipDTO> @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method is called from the service fetchRelationshipsForExistingParty. It fetches the related party details from party direct relationship table. @param SessionContext @param partyIdList

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchAllPartyRelationships		This method returns all the relationships for the party Id @param sessionContext @param partyId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	checkMandatoryRelationshipForPartyType		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBySpidAndRoleId		This method will update the agent details in header if the submission is sourced by an Agent. @param sessionContext @param applicationSearchResponse @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBySpidAndRoleId		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyBySpidAndRoleId		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationshipDetails	@param sessionContext @param relationshipCode @return @throws FatalException	@param submissionInputDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationShipDetails	@param sessionContext @param relationshipCode @return @throws FatalException	This method is used to send alerts to all parties linked with specific party as owner , administrator or others etc. @param applicationContext @param originationActivityDTO @param sessionContext @param partyId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	validatePartyFinancialProfile		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchNoOfAccountsForParty		This method validates that for the party passed and each chosen offer, the number of accounts to be applied plus existing accounts does not exceed the predefined maximum number of accounts per customer for a particular offer.  A query is used to check the total number of accounts that are already opened for the applicant for a given offer whereby the relationship of the applicant to the account is either SOW, JAF or JAO.  Also, accounts that are marked as 'CLOSED' are ignored in the count.   Finally, a check for the total count is checked against the max limit set for each offer in PMU and if it exceeds, an error is thrown and the user is not allowed to setup the account.   Originally, the validation was performed at product level instead of offer level. Additionally, the validation was performed for DDA, OD, TD, LOANS instead of DD and TD. @param sessionContext @param partyId @param productCode @param currencyCode @param accountSerialNumber @param productGroupLinkageType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyNameBySpidAndRoleId		This method will fetch the agent name for the input agent Id. @param roleId @param agentId @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyNameBySpidAndRoleId		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchTemplatePriority		This service is used for resolving the template for financial profile capture. An applicant can apply to multiple product groups within the same submission.   Based on the information provided for the applicant , the product group one applies to and the purpose for application different templates may be resolved.  This information is provided to the rule engine which returns a template identifier per product group.  If an applicant applies to multiple product groups, the priorities for all templates are compared and the one  with the highest priority is stamped against the applicant within the submission.  @param sessionContext @param submissionInputDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchFinancialTemplate		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchFinancialTemplate		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	resolveFinancialTemplateForApplicant		This service is used for resolving the template for financial profile capture. An applicant can apply to multiple product groups within the same submission.   Based on the information provided for the applicant , the product group one applies to and the purpose for application different templates may be resolved.  This information is provided to the rule engine which returns a template identifier per product group.  If an applicant applies to multiple product groups, the priorities for all templates are compared and the one  with the highest priority is stamped against the applicant within the submission.  @param sessionContext @param submissionInputDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	resolveFinancialTemplateForApplicant		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForDebtServicing		Create applicant financial summary by adding financial parameters of the parties in debt servicing group for the selected financial period @param sessionContext @param debtServicingGroupDTO @param currencyCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForDebtServicing		This service will populate financial periods for all parties in the debt servicing group passed in input. The values will be derived from business income statements in party module.i.e it will be populated only for Organizations, Trusts or Self employed individuals. @param sessionContext @param debtServicingGroupDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForDebtServicing		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForDebtServicing		This service will fetch the party 's current financial profile @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchPartyFinancialProfileForDebtServicing		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	calculateNetCustomerPositionDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	createNewParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRelationship		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	generatePartyId		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchSetIdForPartyType		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchApplicableKYCMCode		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	initiateKYC Check		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	fetchRoleConfigurationForRoleId		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	addParty		This method would return {@link ApplicationDecisionDTO}. During the creation of the DTO the factory (instance of {@link ICreditDecisionFactory}) would be invoked for values to be set in the ApplicationDecisionDTO object @return ApplicantDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	addParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	addParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will validate if all the applicants applying for one or more offers are eligible for setting up accounts using these products. Multiple applications contain different kinds of offers. Following validations are performed:</p> <ol style="list-style-type: none"> <li>1. Validate the offer and party based on rule setup in rule engine. These are done at various restriction levels such as Product Group, branch, etc.</li> <li>2. Check for any applicant warning indicators present for account opening events. If present, such applicants should be restricted from being added to accounts.</li> <li>3. Call to check if party is eligible for all offers applied.</li> <li>4. Check for accounts already setup for the party for the same product against maximum number of accounts allowed for the product setup in PMU module. If the maximum limit is exceeded, an error is thrown and the given product cannot be setup for the applicant.</li> </ol> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantsProductValidationDTO, applicantsProductValidationDTO, , Indicates the list of applicants to be evaluated for the products applied for @return ApplicantsRuleRestrictionResponse, , Response object containing the results for the applicants from rule engine @throws FatalException @exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICANT_PRODUCT_INVALID} : FC_OR_DE_0067 &lt;br&gt; This error occurs when:&lt;br&gt; The offers being selected by the applicant cannot be applied due to restrictions setup&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#MAX_ACCTS_EXCEEDED} : FC_OR_DE_0065 &lt;br&gt; This error occurs</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>when:&lt;br&gt; In case of deposit products (CASA and TD), each applicant can apply for limited number of accounts under the product defined in preferences setup in PMU. If the applicant tries to apply for more than the max number of accounts, this error will be seen &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#MAX_ACCTS_REACHED_BRN} : FC_OR_DE_0068 &lt;br&gt; This error occurs when:&lt;br&gt; In case of deposit products (CASA and TD), each applicant can apply for limited number of accounts under the product defined in preferences setup in PMU. If the applicant tries to apply for more than the max number of accounts, this error will be seen. This error message gives more details on which applicant has breached the limit and how many accounts have already been setup &lt;br&gt; Presently, the validation is performed for DD, OD at offer level instead of product level. &lt;br /&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND} : FC_OR_DE_0054 &lt;br&gt; This error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account opening are present for applicants applying to the products.    {@link OriginationBusinessPolicyConstants#APPLICANT_PRODUCTS_BUSINESS_POLICY} : FC_OR_BP_002   This error occurs when:  Applicants are not eligible for applying the selected product groups due to any of the exception stated above. 
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will return facts required for pricing during structure solution in pre assessment for all lending product groups in the submission. Currently total direct credit exposure is being populated. It is computed in LCM for existing parties and proposed and credit facilities.   The inputs will be sent from Origination based on below logic. Consider the following scenario. A is existing party and B is new party.  <table border="1"> <tr> <td><b>Party</b></td> <td><b>Product Group</b></td> <td><b>Requested Amount</b></td> </tr> <tr> <td>A</td> <td>Home Loan</td> <td>100k</td> </tr> <tr> <td>B</td> <td>Personal Loan</td> <td>50k</td> </tr> <tr> <td>A, B</td> <td>Over Draft</td> <td>150k</td> </tr> </table> The following input will be passed to LCM in order to calculate the total direct credit exposure for each product group  <table border="1"> <tr> <td><b>For Product Group</b></td> <td><b>Party</b></td> <td><b>Facility</b></td> </tr> <tr> <td>Home



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Loan&lt;/td&gt; &lt;td&gt;A&lt;/td&gt; &lt;td&gt;Home Loan, Over Draft&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Personal Loan&lt;/td&gt;; &lt;td&gt;B&lt;/td&gt; &lt;td&gt;Personal Loan, Over Draft&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Over Draft&lt;/td&gt; &lt;td&gt;A, B&lt;/td&gt; &lt;td&gt;Home Loan,Personal Loan,Over Draft&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SubmissionInputDTO, submissionInputDTO, , Indicates the draft submission input DTO @return StructureSolutionFactsResponse, An instance of the response containing facts per lending product group required for structure solution. @throws FatalException</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will populate the Applicant Data DTO using the party identifier of the applicant.&lt;br&gt; The party details are fetched using the MDM service. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDTO applicantDTO, , The applicant information for which details are being populated. @return ApplicantDataDTO, Returns applicant information. @throws FatalException</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service returns the applicant to product group linkages. Detailed product group linkage DTO information is available @param applicantDataDTOs @param productDetailsDTOs @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will return a hash map containing all the parties ' id in the application and the corresponding party names. @param applicationId @return HashMap @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to get an instance of the Party Adapter @return IOriginationPartyAdapter @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This adapter call is used to fetch CCI premium amount quote.
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param sessionContext @param accountId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will populate subscription details DTO with subscribing party details for offers linked to bundle. @param productGroupLinkageDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method would return <a href="#">ApplicationDecisionDTO</a> . During the creation of the DTO the factory (instance of <a href="#">ICreditDecisionFactory</a> ) would be invoked for values to be set in the ApplicationDecisionDTO object @return ApplicantDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param submissionInputDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate the Applicant Data DTO from Applicant DTO @param sessionContext @param applicantDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service is used to fetch the eligible offers for the selected product group from PMU module.&lt;br&gt; An eligibility check is run based on application channel, applicants applying for the product group and facility parameters.&lt;br&gt; A list of offers complying with the above conditions are returned. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO[], applicantDataDTOs, , Contains information about applicants applying for the product group. @fcb.param MI, ProductDetailsDTO , productDetailsDTO, , Contains information about product group and facility in case of lending product groups. @fcb.param MI, HeaderDTO, submissionHeaderDTO, , Indicates the branch details, channel details etc for a submission @return OfferInquiryResponse, Returns list of eligible offers for the product group. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#CAMPAIGN_CODE_INVALID} : FC_OR_DE_0085 &lt;br&gt; This exception is thrown if campaign code provided is incorrect or not valid for the product group. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FAILED_IN_RETRIEVING_ELIGIBLE_OFFERS} : FC_OR_DE_0086 &lt;br&gt; This exception is thrown if offers cannot be retrieved for the product group. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_REQUESTED_TERM_NULL} : FC_OR_DE_0101 &lt;br&gt; This exception is thrown if the requested term is null. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service checks if any applicant from the list of applicants is a trust party. @param applicantDataDTOs @return ApplicantDataDTO, Returns the trust applicant if present, or null
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service checks if any applicant from the list of applicants is a corporate trust. @param applicantDataDTOs @return ApplicantDataDTO, Returns the corporate trust applicant if present, or null
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service is used for resolving the template for financial profile capture. An applicant can apply to multiple product groups within the same submission.   Based on the information provided for the applicant , the product group one applies to and the purpose for application different templates may be resolved.  This information is provided to the rule engine which returns a template identifier per product group.  If an applicant applies to multiple product groups, the priorities for all templates are compared and the one  with the highest priority is stamped against the applicant within the submission.  @param sessionContext @param submissionInputDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will execute the rule for Employee benefits if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for Employee benefits rule. @return Boolean, Returns if Employee benefits are applicable or not @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate the fact required for the Low Doc rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details @fcb.param MI, Map<String, CollateralDTO>, collateralMap, , Indicates the collateral details with collateral code as the key. Collaterals linked to the facility are fetched using this collateral code. @fcb.param MI,Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates the map containing applicant data with applicant serial number as the key. @return FactInformationDTO[], Returns a list of facts required for execution of Low Doc Rule. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate the fact required for the Employee rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant for the product group @fcb.param MI,ApplicantRelationshipDTO[], applicantRelationshipDTO, , Indicates the information about relationship between applicants of a submission. @fcb.param MI,Map<Integer, ApplicantDataDTO>applicantMap, , Indicates map of applicant data with serial number as the key. @return FactInformationDTO[], List of facts to be passed for Employee Rule Execution @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will populate reason codes and fields like requested amount & term for facility decision, titleSearchRequired flag and valuation type description for collateral decision and party name for applicant decision. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,ApplicationDecisionDTO, applicationDecisionDTO,, @fcb.param MI,LendingApplication, lendingApplication @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This is called from inquireSubmissionDecision and used to populate amounts in party debt servicing and debt servicing group details @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,DebtServicingGroupDTO,debtServicingGroupDTOS @fcb.param MI,SubmissionDecision,submissionDecision @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to populate the {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} with the creditDecisionId This method also populates the reasonCodeReferenceId inside the CreditReasonCodeReferenceDTO @fcb.param MI,SubmissionDecisionDTO,submissionDecisionDTO, ,SubmissionDecisionDTO containing all the decision details of the whole submission {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} @fcb.param MI,String,creditDecisionId, ,The creditDecisionId to be populated into SubmissionDecisionDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param sessionContext @param applicationId @param borrowingEntity @param applicantDecisionDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch details from LCM and populate proposed conditions and covenants with status as Proposed_New or Proposed_Amend in ConditionItemDetailsDTO array @fcb.param MI,SessionContext,sessionContext @fcb.param MI,SubmissionDecisionDTO,submissionDecisionDTO @returns MI, ConditionItemDetailsDTO[] @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This adapter call is used to fetch CCI premium amount quote.
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service is called from funding table to get LMI quote from the third party provider @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO productGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,ApplicantDataDTO[] applicantDataDTOs , This contains the details of the applicants who   have applied for the product group. @fcb.param MI,CollateralDTO[] collateralDTOs , This contains the collateral details linked to the product group.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will populate the {@link BasicApplicationSearchResponseDTO} with the required attributes from the {@link SubmissionSummarizedEntity}. @param submissionSummarizedEntity @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch the agent name for the input agent Id. @param roleId @param agentId @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to build ProductGroupApprovalDataDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method constructs the TD account opening DTO @param sessionContext @param accountConfigDTO @param applicationId @return TermDepositAccountOpenDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Verify the account limits  1. Each account limit should be less than parent limit.  2. Each account term should be less than parent term.  3. Sum of all accounts under a facility should be less than or equal to parent limit.   4. Sum of all accounts configured under a facility of type non revolving should be less than or   equal to main facility limit and also conform to its own parent limit (same as 3 above).   5. OD Fixed Term should be less than or equal to its parent term.     Also, values for DPA (UDM, fees negotiations are read from each account configuration and set with the highest values. @param sessionContext @param creditAccountDTOs @param facilityId @param parentLimit @param parentTerm @param solutionDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantProductGroupReferenceDTO[] applicantsForPG, , Indicates the applicants of the product group @return List<ValidationError>, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method checks if the KYC has been confirmed for all parties applying to the product group for which account is being opened. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String[] partyIds, , Indicates the party id linked to account @return List<ValidationError>, If KYC is not confirmed for any of the applicants, errors are thrown for the applicant @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to fetch a list of applicants by passing an applicationId. NOTE: Caller must make sure that the input applicationId is validated. The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception     {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052   This error occurs when:  The application identifier is invalid and no application details are found for the same. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This method is used to fetch a list of applicants and affiliates for the input applicationId. &lt;br&gt; NOTE: Caller must make sure that the input applicationId is validated. &lt;br&gt; The Applicant list is fetched from the ApplicantDomainService by passing the applicationId. After that, the relationship of the applicants and the Product group are fetched from the ApplicantProductGroupReferenceDTO. The related affiliates are also added to the list of applicants. Then the ApplicantDTO array is generated from the applicant information. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId, , Indicates the application for which applicants are being fetched. @fcb.param MI,Boolean isPartyDetailsRequired, , Indicates if details party information is to be fetched for the applicants. @return ApplicantDTO[], Indicates the list of applicants applying for the application. @throws FatalException @exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052 &lt;br&gt; This error occurs when:&lt;br&gt; The application identifier is invalid and no application details are found for the same.&lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		update loan simulation with our details
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service is used to fetch the eligible offers for the selected product group from PMU module.  An eligibility check is run based on application channel, applicants applying for the product group and facility parameters.  A list of offers complying with the above conditions are returned. @param sessionContext @param partyId @param productGroupDTO @param submissionHeaderDTO @return OfferInquiryResponse @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param debtServicingGroupDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will verify if same applicant is present in more than one debt servicing group @param debtServicingGroupDTOs
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will compute and return the debt servicing financial summary for the debt servicing group passed in input. The financial details are computed based on debt servicing financial period selected by user. In case of the first call after debt servicing group creation, financial period will be defaulted from applicant income details returned by decision engine for parties in the group. In all subsequent calls, the financial period is mandatory. The applicant financial summary is derived by adding financial parameters of the parties in debt servicing group for the selected financial period. Debt servicing financial details is derived by adding financial parameters for all parties for the selected financial period in the debt servicing group @param com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param debtServicingGroupDTO @return DebtServicingGroupInquiryResponse @exception {@link OriginationErrorConstants#OR_INV_INPUT} : FC_OR_DE_0019  This exception is thrown when the service is called without debt servicing financial

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				period and debt servicing financial details are present  @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Create applicant financial summary by adding financial parameters of the parties in debt servicing group for the selected financial period @param sessionContext @param debtServicingGroupDTO @param currencyCode @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param applicantFinancialDetail
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate financial periods for all parties in the debt servicing group passed in input. The values will be derived from business income statements in party module.i.e it will be populated only for Organizations, Trusts or Self employed individuals. @param sessionContext @param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				debtServicingGroupDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will call LCM service to save the MANUAL condition linkages entered by user from application form as well as condition linkages returned by decision engine.   Before calling LCM the manual conditions are filtered to remove duplicate combinations for BE + Entity Id + Condition Code across the applications.   Origination will save the condition linkage id in ApplicationDecisionConditionLinkage. @param sessionContext @param submissionDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service fetches party details and populates factsMap to execute condition/covenant policy @param sessionContext @param applicantDecisionDTO @param borrowingEntityId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<<2.3>>  This method is used to retrieve and save the PartyConductDetailsDTOs for one given ApplicationDecisionDTO @param applicationDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to inquire the partyConductDetailsDTOs and populate them into the ApplicationDecisionDTO @param applicationDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will populate the isShareIncomeFlag and isShareExposureFlag of the applicant decision dto <1> Main applicant, both flag populated as true <2> Direct affiliate, populated based on the flag value in the party to party relationship <3> Indirect affiliate, populated as true only when the flag value of itself and its upper level affiliate are true @param sessionContext @param submissionDTO @param submissionDecisionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to log an event/alert for KnowYourCustomer. The method calls publishOriginationActivity from OriginationEventApplicationService which handles and processes events. It will internally call registerActivity from AlertsNotificationHandler which performs eventProcessing. @param sessionContext @param submissionId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Added method to send alert to primary applicant of Product group under each application in submission. This method will only be called when submission goes for manual review in either case for credit   stage of submission method wont be called. @param sessionContext @param submissionDecisionDTO @param eventType
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates an array of {@link ApplicantDataDTO} which contains information about the applicants which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @return an array of {@link ApplicantDataDTO} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch the party to party relationships between all parties for the given borrowing entity id @param sessionContext the {@link SessionContext} @param borrowingEntityId the borrowing entity id @return an array of {@link PartyToPartyRelationshipDTO} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will link the applicants to the product group to which they were linked at the time of submission. @param sessionContext the {@link SessionContext} @param accountCode the Account Id @param applicantDataList the List of Applicants @param productGroupLinkageDTO the Product Group Linkage @return the {@link ProductGroupLinkageDTO} with applicants linked @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a list {@link ApplicantDataDTO} by extracting information from the given {@link MDMPartyDTO} @param mdmPartyDTO the {@link MDMPartyDTO} @return a list of {@link ApplicantDataDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a new {@link ApplicantDataDTO} and sets its appropriate attributes with the information extracted from the given {@link IndividualDTO} @param individualDTO the given {@link IndividualDTO} @return the {@link ApplicantDataDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a new {@link ApplicantDataDTO} and sets its appropriate attributes with the information extracted from the given {@link OrganizationDTO} @param organizationDTO the given {@link OrganizationDTO} @return the {@link ApplicantDataDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a new {@link ApplicantDataDTO} and sets its appropriate attributes with the information extracted from the given {@link TrustDTO} @param trustDTO the given {@link TrustDTO} @return the {@link ApplicantDataDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a {@link PartyOnBoardingDTO} with {@link PartyType#Individual} from the given {@link IndividualDTO} @param individualDTO the given {@link IndividualDTO} @return the {@link PartyOnBoardingDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a {@link PartyOnBoardingDTO} with {@link PartyType#Legal_entity} from the given {@link OrganizationDTO} @param organizationDTO the given {@link OrganizationDTO} @return the {@link PartyOnBoardingDTO}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method creates a {@link PartyOnBoardingDTO} with {@link PartyType#Trust} from the given {@link TrustDTO} @param trustDTO the given {@link TrustDTO} @return the {@link PartyOnBoardingDTO}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method would validate whether an individual is linked to P2A. It performs the following operations <ul style="list-style-type: none"> <li>&lt;br&gt; &lt;li&gt;Create a List of main-applicants from P2A&lt;/li&gt;&lt;br&gt; &lt;li&gt;Create a List of affiliates from P2A&lt;/li&gt; &lt;br&gt; &lt;li&gt;Create a List of main applicants with ApplicantRelationshipType not GUARANTORS and ApplicantStatus not APPLICANT_INFO_NOT_AVAILABLE&lt;/li&gt;&lt;br&gt; &lt;li&gt;Validate that all the main applicants from &lt;i&gt;applicantPartyIds&lt;/i&gt; are present in the P2A &lt;i&gt;partyToAccRelPartyIds&lt;/i&gt; and vice versa&lt;/li&gt;&lt;br&gt; &lt;li&gt;Validate that the P2A affiliates should be present in the applicant data list for the submission&lt;/li&gt; @param partyToAccountRelationshipDTOList @param productGroupLinkageDTO @param submissionInputDTO @return</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method would validate whether an individual is linked to P2A. It performs the following operations   <li>Create a List of main-applicants from P2A</li>  <li>Create a List of affiliates from P2A</li>   <li>Create a List to maintain all the Individual applicants</li>  <li>Check whether at-least one individual is linked to P2A</li>  @param partyToAccountRelationshipDTOList @param productGroupLinkageDTO @param submissionInputDTO @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method iterates over the list of collateral attached for the lending product group and checks the number of collateral of type GUARANTEE. It returns the number of collateral of GUARANTEE type. @param collateralDTOs @param collateralCodes @return {@link Integer}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method checks that the passed parties are granted only predefined permissible roles. It makes use of party module - RoleConfigurationApplicationService.fetchRoleConfiguration. @param sessionContext @param applicantDataListDTOs @return List<ValidationError>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to send alerts to all applicants present at time of account opening.  For each applicant we check the status of Applicant to see if applicant was present , if   Applicant Status is marked as APPLICANT_INFO_NOT_AVAILABLE the applicant will enter the   looping logic. @param sessionContext @param originationActivityDTO @param applicationContext @param applicantsList
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch all applicants for application ID received in inputs. @param sessionContext @param originationActivityDTO @param applicants @param applicationContext
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch all applicants under given submission ID. @param sessionContext @param originationActivityDTO @param applicants @param applicationContext
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method takes all the applicant details as arguments and passes them to register activity method to log the alert @param loanDTO @param sessionContext @param applicants @param applicationContext



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This alerts will be send to all applicants during account opening for CASA and TD applications. Alert will be send to all parties linked to account. While sending alerts to condition will be checked. <ol> <li>If PartyAccountRelationshipType is not PartyAccountRelationshipType.OTHERS send alert will be send to that party.</li> <li>If PartyAccountRelationshipType is PartyAccountRelationshipType.OTHERS then alert will be send only if for given affiliate alert to be send is allowed or not.</li> </ol> @param sessionContext @param originationActivityDTO @param applicationContext
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This function is used to send alerts for KYC Pending in case of Trust and Corporate case. In this case individual affiliates linked to this organisation/trust and linked to account while creation will get alert. @param sessionContext @param originationActivityDTO @param accountHolders @param partyId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to send alerts to all parties linked with specific party as owner , administrator or others etc. @param applicationContext @param originationActivityDTO @param sessionContext @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Check if KYC is not done for any party @param sessionContext @param applicantDTOs @return boolean @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Fetch unique party ids for product group @param applicantProductGroupReferences @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to populated header information into application basic info DTO @param applicationBasicInfoDTO @param headerDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details.   The submission draft can be retrieved in the following ways:   1. My Tracker - This tracker is used by the applicant to log into the system.   2. Application Tracker - This tracker is used by the banking associated to log into the system.   When the submission is saved, all applicants are on-boarded as 'Member' in the party system. The process takes care of on-boarding process  The applicants can later log into the system and submit the application or the banker can retrieve the submission using the applicants and continue with the submission.  The inputs submissionInputDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>will be serialized into String and passed to DraftApplicationService for save as draft. &lt;br&gt;This service is also called by BPEL process to save the draft submissionInputDTO in the following scenarios-&lt;br&gt;i) before calling createSubmission &lt;br&gt; ii) if any error is encountered on confirm of submission.&lt;br&gt;@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionProcessDraftDTO, ,This class holds the entire information of one submission and contains the input details as well as the processed output from the process @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will create a Submission from the input details .&lt;br/&gt; &lt;br/&gt;The service will be called from the BPEL process. The following activities will be carried out in sequence.&lt;br/&gt; &lt;br/&gt; a) The service creates all new applicants for this submission by iterating through the list of ApplicantData.&lt;br/&gt; &lt;br/&gt; b) This service will create applications based on the Product details submitted by the user.&lt;br&gt; The logic of creating multiple applications is as follows:&lt;br&gt; &lt;br&gt; 1) All deposit product groups applied will be put into separate applications.&lt;br&gt; 2) All lending product groups newly applied will be split into separate applications based on the borrowing entity (group of parties identified by borrowing entity identifier). Different product groups applied by the same borrowing entity will be grouped under the same application.This is applicable for amended facilities as well.&lt;br&gt; 3) All lending product groups that are applied as a result of amending existing facilities will be put into separate applications. &lt;br&gt; &lt;br&gt; c) Simulation details are saved using the corresponding simulation application services based on the product group. &lt;br&gt; For e.g.&lt;br&gt; Demand Deposit Product Simulation will be saved using the DemandDepositSimulationApplication Service. &lt;br/&gt; Term Deposit Product Simulation will be saved using the TermDepositSimulationApplication Service. &lt;br/&gt; Overdraft Product Simulation will be saved using the DemandDepositSimulationApplication Service. &lt;br/&gt; &lt;br&gt; d) The submission status is marked as SUBMISSION_SUBMITTED.&lt;br&gt; &lt;br&gt; e) While creating a submission, the history records for submission are checked. If a record already exists, the number of history records for that submission identifier are set. This is required as an input to credit</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>decision engine. &lt;br&gt; &lt;br&gt; f) Submission expiry Date is calculated and saved using preference based on Brand and Market Entity.&lt;br&gt; &lt;br&gt; g) For new applicants with generated party identifiers, the new identifiers are replaced in the Inbound Document Checklist.&lt;br&gt; h) In case of bundle applied within the submission, bundle subscription is done for all owners of the submission.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse} This class holds the result of the service. An instance of this class holds a reference to an instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed and an instance of the SubmissionDTO which contains all information of the created submission. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInquiryResponse} for more details. @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> <a href="#">FatalException @exception</a> &lt;br&gt; <a href="#">{@link OriginationErrorConstants#SUBMISSION_NOT_FOUND}</a> : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; <a href="#">{@link OriginationErrorConstants#OR_INV_OUTPUT}</a> : FC_OR_DE_0064 &lt;br&gt; This error occurs when:&lt;br&gt; The details for applicants that have been created or updated does not match the input list of applicants data&lt;br&gt; &lt;br&gt; <a href="#">{@link OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND}</a>&lt;br&gt; FC_OR_DE_0075 error occurs when:&lt;br&gt; The submission history for the submission cannot be fetched.&lt;br&gt; &lt;br&gt; </p>
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p> This method updates all affiliates' documents - owner id and document indexes - index values with the appropriate party id. Originally, for indexes - index values, the party id recorded is submissionId + underscore + partySerialNumber. @param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException </p>
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p> @param sessionContext @param submissionInputDTO @param applicantMap @throws FatalException </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to update the submission summarized entity with the application ids, party ids, submission type and total request / approved amount which are not part of the Submission entity persistence @param submissionId, {@link String} @param existingSubmissionDTO (@link SubmissionDTO) @param applicantDTOs @return void
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will create the party names string for the submission. @param submissionInputDTO @param applicantDTOs @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will create new applicants or update existing applicants for a submission. Only applicants for which party id is generated or whose status is APPLICANT_INFO_NOT_AVAILABLE are persisted. This method will be called on submission draft save as well as on create submission. @param sessionContext @param submissionInputDTO @param existingSubmissionDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will set the submission id in applicant offer linkages.  It also decides the primary applicant for the submission if not already set.  If applying party is present and is an Individual, the applicant corresponding   to the applying party is marked as primary applicant for submission.  If applying party is present and is a Trust or Corporate party, the first individual  amongst the list of applicants is marked as Primary applicant.  If no applying party is present, the first Individual is marked as Primary Applicant.  This case arises when all applicants applying are NEW parties and onboarded as part of this submission.  @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				applicantDTOs @param existingSubmissionDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will find the ApplicantDataDTO for the given Applicant serial number from the list. Then it will return the PartyId of the applicant if its found, otherwise it will return null. @param applicantDataDTOList @param applicantSerialNumber @return
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate facts for applicant , product group and collateral (optional) required for Inbound documentation. This information is passed to the Document Management System which evaluates the facts for each applicant for all product groups applied and collateral(s) if submitted for the applicant. It internally creates a set of document identifiers to identify the documents both mandatory and optional to be provided by the applicant. All document identifiers are grouped under a single Document Checklist and the identifier for this checklist is provided to origination. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.Submi



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ssionInputDTO} for more details. @return DocumentChecklistInputResponse, Contains the input dto for inbound document checklist creation. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#FAILED_IN_GEN_INBOUND_CHECKLIST} : FC_OR_DE_0087 &lt;br&gt; This exception is thrown if there is a problem in inbound document checklist generation in DMS. &lt;br&gt; &lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This method populates the Document Checklist Record Input DTO for applicant - product group linkage. @param sessionContext @param submissionId @param applicantMap @param productGroupLinkageDTO @return @throws FatalException</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This method populates the Document Checklist Record Input DTO for affiliate - product group linkage. @param sessionContext @param submissionId @param applicantSerialNumber @param applicantMap @param partyToPartyRelationshipDTOs @param productGroupLinkageDTO @param eventId @return @throws FatalException</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service checks if the party id exists for an applicant. If party Id is available, the same is saved to DMS else applicant serial number. @param documentChecklistRecordInputDTO @param applicantDataDTO @param submissionId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will create a simulation or save an existing simulation for overdraft products being setup for an application. Each overdraft product group is attached to a facility maintained in LCM system. In case of overdraft product groups, the user can setup multiple simulations for different offers selected. Each offer corresponds to a single simulation and contains the account configuration and additional details. The primary applicant will provide various preference details such as cheque book preference, deposit book preference, etc for the account. Preferences are also captured for each applicant of the account for statement, debit card details, etc. Overdraft facility can also be opted for existing demand deposit accounts in which case the user does not have to provide the configuration details since the account is already active. An existing demand deposit account can be attached only if the following conditions are fulfilled: &lt;br&gt; &lt;br&gt; Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners. &lt;br&gt; &lt;b&gt;For e.g.&lt;/b&gt; : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list. &lt;br&gt; Existing CASA account selected for Overdraft, must not be linked to any other Overdraft facility. &lt;br&gt; &lt;br&gt; The system will internally create a linkage for this simulation to be fetched for future retrieval. The applicant(s) can continue making changes to this simulation until the account is opened. @fcb.param MI,SessionContext,sessionContext, ,Session Context</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String submissionId, , Indicates the identifier of the submission for which the the simulation details are being saved @fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application. @fcb.param MI,ODSimulationDTO, odSimulationDTO, , This class holds simulation details for the overdraft product. It contains information about the accounts being configured within the simulation such as statement preferences, card preferences, etc. @return {@link ODSimulationResponse} This class contains the simulation details that passed and further updated within this service. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method validates whether the parties present in the submission has any warning indicators applied. If there is any warning indicators applicable for any of the party then the system should not create the application and also add a proper validation error with an appropriate message. @param sessionContext @param submissionInputDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method populates the map between applicantDTO and Party id
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method iterates over the list of collateral attached for the lending product group and checks the number of collateral of type GUARANTEE. It returns the number of collateral of GUARANTEE type. @param collateralDTOs @param collateralCodes @return {@link Integer}
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Validate if offer already applied and not allowed to apply again. 2.2: Applicable for only Credit Card offers. <ul> <li>Credit Card offer validation: Check if account already opened / any application is in progress for the party id, offer id combination</li> </ul> @param sessionContext @param partyId @param offerId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will construct facts for the application , these facts are to be passed to the DMS to get the list of offer documents for the application. @fcb.param MI, LendingApplication - Application details which will be used to construct the factsDTO. @fcb.param MI, DecisionStatusType - Credit decision for the application @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @return DocumentPolicyDetailDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will retrieve the list of collateral owners and pass it for the construction of RelationshipDTO. This list of collateral owners is added to the existing relationship list. @param sessionContext @param appPgDTOList @param ownership @return RelationshipDTO[] @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will construct the relationship for the application, the delivery preferences are set using the relationship. @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		The method will construct facts for the primary applicant in the application @fcb.param MI, applicationId - Application details which will be used to construct the factsDTO. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		The method will construct facts for the collaterals in the application @fcb.param MI, LendingApplication - Application details which will be used to construct the factsDTO. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to fetch relationship DTO for related parties with class, Owner / Administrator and relation type, Business. @param sessionContext @param partyId @return List<RelationshipDTO> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to construct Document policy for parties linked to all lending product group linkage. The linkage type / level is Party. The details send for the policy execution are, <ul> Facts <li>Party (MDM) facts</li> <li>CCI facts</li> <li>Lending product group linkage facts</li> </ul> <ul> Report Request DTO / Auxiliary Linkage DTO params <li>Party Id</li> <li>Application Id</li> <li>Submission Id</li> <li>Product Group Id</li> <li>Product Group Serial Number</li> <li>Facility Id</li> </ul> <ul> Relationship DTOs <li>Applicant / Co-Applicant / Guarantor / Solicitor</li> <li>Corresponding Affiliates (Class: owner / administrator, Relation type: business)</li> </ul> @param sessionContext @param lendingApplicationDTO @param serviceEventId @param submissionId @return List<DocumentPolicyDetailDTO> @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Construct facts for part from Origination fact application (MDM party facts) @param partyFacts @param applicantId @param partyId @param sessionContext @return String @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to construct Relationship DTO for party. <ul> Relationship DTOs <li>Self</li> <li>Corresponding Affiliates (Class: owner / administrator, Relation type: business)</li> </ul> @param sessionContext @param applicantProductGroupReferenceDTO @return RelationshipDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility.&lt;br&gt; An existing demand deposit account can be used only if the following conditions are fulfilled: &lt;br&gt; &lt;br&gt; Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners. &lt;br&gt; &lt;br&gt; &lt;b&gt;For e.g.&lt;/b&gt; : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list. &lt;br&gt; &lt;br&gt; Also, existing CASA selected, must not be linked to any other Overdraft facility. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @fcb.param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#OR_OFFER_INVALID} : FC_OR_DE_0081 &lt;br&gt; This error occurs when:&lt;br&gt; The offer details are being fetched for incorrect offer identifier or offer has expired.&lt;br&gt; &lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method performs the following validations: 1. There should not be any warning indicators for the parties applying for the PG 2. The KYC check should be completed for all the parties 3. All the preliminary conditions should have been complied with 4. Productgroup level Origination fees should have been collected 5. Application should be verified successfully @param sessionContext @param application @param prodGrpLinkage @return @throws Exception
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to get an instance of the Party Adapter @return IOriginationApplicantAdapter @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will check null inputs for create or updating facility. @param submissionInputDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will confirm structure solution for lending product group in a submission. &lt;br&gt; Various operations performed during this service are as follows: &lt;br&gt; &lt;ol&gt; &lt;li&gt;The facility structure is validated as per the guidelines. {@link LendingApplicationHelper#checkFacilityConformity}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Validations for confirming structure solution are validated. {@link LendingApplicationHelper#validateConfirmActionOnStructureSolution}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if ticket number for Origination fees is present. If yes, defer all the uncollected fees for that ticket number else create a new pricing plan, fetch and defer the origination fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if product group is setup under a bundle and ticket number for bundle fees is present. If yes, defer all the fees for that ticket number else create a new pricing plan, fetch and defer the bundle fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;In case of Amended facility, check if settlement is required or not (irrespective of the settlement flag on product group and update the same) {@link LendingApplicationHelper#checkIfSettlementRequiredForPG}&lt;/li&gt;&lt;br&gt; &lt;li&gt;For Loan product groups, update party id, borrowing entity and facility details for loan simulations&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check for any rates and fee negotiations. If present, send for discretionary pricing approval&lt;/li&gt;&lt;br&gt; &lt;/ol&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @return LendingAccountCreationResponse, this class returns</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the LendingAccountCreationResponse object,  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#KYC_DETAILS_NOT_FO  UND} : FC_OR_PI_0001 &lt;br&gt; This check is part of  Pre-create account validation. &lt;br&gt; This exception is  thrown if KYC has not been confirmed for all parties  &lt;br&gt; applying to the product group for which account  is being opened. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_DPA_APPROVAL} :  FC_OR_DE_0082 &lt;br&gt; This exception is raised as an  overridable exception when pricing rates and fees are  negotiated and require approval. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_AMOUNT_  GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception  occurs when one or more sub facilities are configured  and the requested amount for these sub-lines exceed  their parent facility's (can be main facility or another  sub facility) amount. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_TERM_GRE  ATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs  when one or more sub facilities are configured and the  requested term for these sub-lines exceed their parent  facility's (can be main facility or another sub facility)  term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_  SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#LVR_THRESHOLD_BREA  CHED} : FC_OR_SS_0006 &lt;br&gt; This exception occurs  when the loan to value ratio is breached beyond the  limit allowed for a product group. This threshold  value is maintained in PMU in credit policy template</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>for the product group &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the a single facility or sub-facility is greater than requested amount of that facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER_THAN_MAIN_FACILITY} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the facility structure exceed the total amount of main facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#ACCT_NOT_CONFIGURED} : FC_OR_SS_0015 &lt;br&gt; This exception occurs when structure solution is being confirmed with no account configured under the facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FETCH_PRICE_OFFERINGS_FAILED} : FC_OR_SS_0016 &lt;br&gt; This exception occurs when fetching of price recommendations for origination fees from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PPC_CREATION_FAILED} : FC_OR_SS_0017 &lt;br&gt; This exception occurs when creation of pricing plan for origination fees in pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#GET_PRICE_ON_EVENT_FAILED} : FC_OR_SS_0018 &lt;br&gt; This exception occurs when fetching price for events in origination (OR_NEW_FACILITY or OR_AMEND_FACILITY) from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OD_FIXED_TERM_GREATER} : FC_OR_SS_0019 &lt;br&gt; This exception occurs when fixed term entered for overdraft product groups</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>exceed the parent requested limit term.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICANTS_HOLDER_PREFERENCES_INCORRECT} : FC_OR_SS_0021&lt;br&gt; This exception occurs when user configures the account configuration in structure solution and then goes back to the product details page and changes the applicants applying for the product group.&lt;br&gt; The user is prompted to check the account configuration again to change the account holder preference for changed applicants.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_FEE_NEGOTIATION} : FC_PR_FEE_OVR &lt;br&gt; This exception occurs any of the rates or fees are negotiated.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#COLLATERAL_VALUATION_INCOMPLETE} : FC_OR_CL_0001 &lt;br&gt; This exception occurs if NCCP is yes and valuation for any collaterals linked to the facility is not yet complete.&lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service validates each sub facility amount is equal to or less than the main facility @param sessionContext @param creditFacilityDTOs @return @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#REQUESTED_AMOUNT_GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception occurs when one or more sub facilities are configured and the requested amount for these sub-lines exceed their parent facility's (can be main facility or another sub facility) amount. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#REQUESTED_TERM_GREATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs when one or more sub facilities are configured and the requested term for these sub-lines exceed their parent facility's (can be main facility or another sub facility)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_  SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#SUM_OF_ALL_ACCTS_G  REATER} : FC_OR_SS_0007 &lt;br&gt; This exception  occurs when the sum of requested amounts of all  accounts under the a single facility or sub-facility is  greater than requested amount of that facility. &lt;br&gt;  &lt;br&gt; {@link  OriginationErrorConstants#SUM_OF_ALL_ACCTS_G  REATER_THAN_MAIN_FACILITY} : FC_OR_SS_0007  &lt;br&gt; This exception occurs when the sum of  requested amounts of all accounts under the facility  structure exceed the total amount of main facility.  &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#ACCT_NOT_CONFIGUR  ED} : FC_OR_SS_0015 &lt;br&gt; This exception occurs  when structure solution is being confirmed with no  account configured under the facility. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will fetch the guarantors for a particular facility. It will check with LCM for the guarantee collaterals linked to this facility and then for all those owners fetch the party details from MDM @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcbparam MI, String, facilityId, , Indicates the identifier for the facility for which hierarchy is being fetched. @return MDMPartiesInquiryResponse , Response containing details of all guarantors for the facility. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This is alert send for Lending Account Creation. For account, it will take corresponding simulationID and fetch the LoanAccountConfigurationDTO. <ol><li>If PartyAccountRelationshipType is not PartyAccountRelationshipType.OTHERS send alert will be send to that party.</li> <li>If PartyAccountRelationshipType is PartyAccountRelationshipType.OTHERS then alert will be send only if for given affiliate alert to be send is allowed or not.</li> </ol> @param sessionContext @param accountOpeningDetailsDTO @param prodGroupSerialNumber @param applicationId
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method populates the lendingProductGroupLinkageSummaryDTO given in input with the fields in lendingproductGroupLinkageDTO which are common for all lending products. @param productGroupLinkage @param productGroupLinkageSummary @return LendingProductGroupLinkageSummaryDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method populates the collateral summary in CollateralLinkageSummaryDTO. @param collateralDTO @return CollateralLinkageSummaryDTO
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOriginationApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Find the primary applicant of a productGroup
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is used to create CCI policy linkages for applicant applied for Consumer Credit Insurance in specific product group. @param sessionContext, It contains contain related to User,Bank Code , Branch etc. @param productGroupLinkageDTO, Represents details to specific product group. @return CCIDetailsResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will be called from UI to fetch Basic CCI Details for Applicant to be shown in CCI Human task as grid. @param sessionContext, contains user information. @param applicationId, denotes the application ID @param productGroupSerialNumber, denotes product group serial number @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.)
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method called from create lending application to persist the CCI quote details. At Funds table in application form, we generate an insurance id, fetch the CCI quote and persist the quote-facility linkage in insurance. The insurance Id is stored in CCIDetailDTO. On submit, if insurance id is present, we delete the existing quote-facility linkage and add quote-facility and quote-party linkages. If insurance id is not present, we generate the insurance id and create insurance-facility and insurance-party linkage. There would be no CCI quote data in this case. @param sessionContext @param productGroupLinkageDTO @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This will fetch all CCI details available under given application. @param sessionContext, contains user information. @param applicationId, denotes the application ID @throws FatalException @link Exception BusinessException OR_INV_INPUT: This exception will be thrown if applicationID is null.)
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is called from the service fetchRelationshipsForExistingParty. It fetches the related party details from party direct relationship table. @param SessionContext @param partyIdList

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method populates the basic party details of related parties found from both LCM and Party services. It populates the basic details and store it in the {@link RelatedPartyInformationDTO} @param sessionContext @param relatedPartyInformationDTOArray @param partyRelationshipList @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service returns the list of parties that are account holders of the account for which the identifier is passed. @param sessionContext @param accountId @return @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method links the Associated Parties created at the submission level to the account level. @param sessionContext @param submissionId @param accountId @param accountType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service will create a term deposit account for given account configuration. It is wrapper service called from process. It does the following in sequence</p> <ol style="list-style-type: none"> <li>1. Calls the TermDepositAccountApplicationService to - populate the TD account opening DTO - open a TD account The account id is set in the response.</li> <li>2. Adds the account reference to the corresponding offer in the application in origination.</li> <li>3. Performs state transition for the product group and the application, if necessary</li> </ol> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p>@fcb.param MI,String, submissionId, , Indicates the identifier for the submission</p> <p>@fcb.param MI,String, applicationId, , Indicates the identifier for the application</p> <p>@fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application.</p> <p>@fcb.param MI, int, offerSerialNumber, , Indicates the offer serial number</p> <p>@fcb.param MI,TDAccountConfigurationDTO, tdAccountConfigurationDTO, , This class holds account configuration details for the term deposit offer. It contains information about the accounts being configured within the simulation such as payin details, payout etc.</p> <p>@return {@link TermDepositAccountResponse}</p> <p>This class contains the account opening response i.e. account id</p> <p>@exception &lt;br&gt; {@link OriginationErrorConstants#OR_DEPOSIT_APPLICATION_INVALID} :FC_OR_ST_0002 &lt;br&gt; This exception occurs during invalid state transition where the application is not a Deposit application &lt;br&gt; &lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service find out MinorAccount status of an account using the date of birth of the primary applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and flgoperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response.</p> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074 &lt;br&gt; This error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089 &lt;br&gt; This error occurs when:&lt;br&gt; The birthdate for the primary applicant cannot be retrieved.&lt;br&gt; &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		<p>This service find out MinorAccount status of an account using the date of birth of the applicant. This will call a utility service in party which will return the status as defined in partyconstants and the fllegalmajor and floperationalmajor. Using these MinorAccountStatus will be then derived for DD and TD account respectively and set in the response.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, primaryApplicantData this contains the primary applicant information. @return ApplicationCreationResponse, , The response contains unique identifiers for each application @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#PRIMARY_APPLICANT_MISSING_SUBMISSION} : FC_OR_DE_0074 &lt;br&gt; This error occurs when:&lt;br&gt; The primary applicant information for which minor status is being fetched is not found.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#BIRTHDATE_NOTAVAILABLE} : FC_OR_DE_0089 &lt;br&gt; This error occurs when:&lt;br&gt; The birthdate for the primary applicant cannot be retrieved.&lt;br&gt; &lt;br&gt;</p>
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		@param partyDataDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method populates the depositProductGroupLinkageSummaryDTO given in input with the fields in depositproductGroupLinkageDTO which are common for all deposit products. @param productGroupLinkage @param productGroupLinkageSummary @return DepositProductGroupLinkageSummaryDTO
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will update the party Id references using the Applicant Data DTO. In case of new parties onboarded, the applicant data dto is updated with the new party identifiers.  All other DTOs requiring party Ids are initially updated using applicant serial number when party Ids are not available. After onboarding process, the applicant serial numbers are replaced with the party Id from Applicant Data DTO. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc.It consists of header information such as application branch, channel, creation date, etc. , details of applicants applying for different product groups, collateral information in case of secured lending product groups. Relationships within the applicants for a submission are also present if provided. @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service updates the party Id in product group linkages. @param linkageDTO @param applicantMap @param applicantSerialNumber @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Common method to repopulate accountAddressLinkageDTO with party id @param applicantMap : serial number / party id
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service checks if the applicants applying to a product group is a trust applicant and returns the Applicant Data DTO for trust. @fcb.param MI, ApplicantProductGroupReferenceDTO[], applicantProductGroupReferenceDTOs, , Indicates the linkage between applicants and product groups i.e. list of applicants applying for a product group. @fcb.param MI, Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates a map of applicant data with applicant serial number as the key @return ApplicantDataDTO, Corresponding to trust party is applying to the given product group else null.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service checks if the applicants applying to a product group is a corporate applicant and returns the Applicant Data DTO for corporate. @fcb.param MI, ApplicantProductGroupReferenceDTO[], applicantProductGroupReferenceDTOs, , Indicates the linkage between applicants and product groups i.e. list of applicants applying for a product group. @fcb.param MI, Map<Integer, ApplicantDataDTO>, applicantMap, , Indicates a map of applicant data with applicant serial number as the key @return ApplicantDataDTO, Corresponding to corporate party is applying to the given product group else null.
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate the party identifier in operating instructions DTO. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the applicant information @fcb.param MI, OperatingInstructionDTO, instructionDTO, , Indicates the operating instructions for which party id is being
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will populate the party identifier in operating instructions DTO. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the applicant information @fcb.param MI, OperatingInstructionsDTO, instructionsDTO, , Indicates the operating instructions for which party id is being updated
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Updates the party id and applicant id in the following places in the DepositAccountHolderPreferenceDTO <li> PartyAddressLinkageDTO.partyId <li>applicantId <li>PartyAccountRelationshipDTO.PartyAccountRelationshipKeyDTO @param accountHolderPreference @param applicantDataDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		Update the party Id in account holder preferences. @param accountHolderPreference @param applicantMap
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service returns the applicant name. In case of Party Onboarding DTO present, this service returns the full name of the party based on the party type.  In other cases, it returns the applicant first name <SPACE> applicant last name as entered on the form. @param applicantDataDTO @return String
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will create default account holder preferences for Corporate and Trust Parties.  This is required for adding party to account relationships for corporate and trust parties.  @fcb.param MI,Set<IApplicantProductGroupReference> applicantProductGroupReferences, , Indicates the applicants of the product group. @fcb.param MI,DepositAccountConfigurationDTO depositAccountConfigurationDTO, , Indicates the deposit account configuration details. @fcb.param MI,String offerName, , Indicates the offer name @return List<DepositAccountHolderPreferenceDTO>, Returns the updates deposit account holder preferences for the account configuration. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service populate the default account holder attributes for non-individual applicants and affiliates linked with the account. @param accountHolderPreferenceDTO @param relationshipDTO @param offerName
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will update the details for the applicant passed. @param sessionContext Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param applicantDTO Indicates the applicant data that is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception   {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0058   This error occurs when:  For the party identifier passed , no details of party are found.    {@link OriginationErrorConstants#PARTY_TYPE_MISMATCH} : FC_OR_DE_0001  This error occurs when:  The party type of the applicant passed

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				does not match the party type stored in Party module. 
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will validate the ApplicantDTO input for the following cases.  <li>1. The applicant should be an existing party</li> <li>2. The existing applicantDTO should have a valid party Id</li> @param sessionContext @param applicantDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will validate data for secondary applicant. @param sessionContext @param applicantDataDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will check the {@link ApplicantDTO} for the following cases  <li>1. Checks whether {@link ApplicantDTO} array is null</li> <li>2. Checks whether Individual {@link ApplicantDTO} is null</li> <li>3. Checks whether PartyType is either Null or empty</li> <li>3. Checks whether First Name is either Null or empty</li> <li>3. Checks whether Last Name is either Null or empty</li> <li> 3. Checks whether Email Id is either Null or empty</li> @param applicantDTOArray @param message1 @param message2
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is to validate the party information for an ApplicantDTO object. For a new applicant who is an EXISTING_PARTY, we need to validate the party_id. And, if the other information is not input, we need to populate them. @param sessionContext @param applicantDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This method is to validate the party information for an ApplicantDTO object. For a new applicant who is an EXISTING_PARTY, we need to validate the party_id. And, if the other information is not input, we need to populate them. This method input ApplicantDataDTO @param sessionContext @param applicantDataDTO @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will update the account holder preferences provided by the secondary applicant for all the accounts setup as a part of the submission and also update the Applicant Status.   a) Prior to account opening, account configuration details are saved in Draft. Information stored in Draft is used for account opening process. For secondary applicants not available at the time of account opening, account holder preferences can be captured post account opening. These preferences are further updated in the draft that is stored for the account configuration.  b) It will update the applicant status as data entry complete.  c) If all the applicants have provided their preferences, then the application status is marked as APPLICATION_PROCESSED and the submission status is marked as SUBMISSION_COMPLETED.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SecondaryApplicantInputDTO, secondaryApplicantInputDTO, , This DTO contains



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				the secondary applicant details that are captured post account opening. Account holder preferences are extracted from this DTO and updated in the draft stored for the account configuration @return An instance of {@link com.ofss.fc.service.response.TransactionStatus} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getParty		This service will called before 'createCustomer' , it will fetch the applicant-data-dto for a given SubmissionId and applicantSerialNumber. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param @fcb.param MI,String,submissionId, ,The submission-id for which the applicant information is to be updated @fcb.param @fcb.param MI,Integer,applicantSerialNumber, ,The applicant serial number for which the applicant information is to be fetched @return An instance of {@link com.ofss.fc.app.origination.dto.core.application.ApplicantInquiryResponse} @throws FatalException
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	getPartyToPartyRelationships		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removePartyToPartyRelationship		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removePartyToPartyRelationship		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	inquirePartyFromDraft		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	inquirePartyFromDraft		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removeParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removeParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removeParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	removePartyToPartyRelationshipForParty		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	addPartyToPartyRelationship		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	isPartyPresentInDraft		
IOrigination ApplicantAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationApplicantAdapter	addAddOnCardHolder		
IPMUPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IPMUPricingAdapter	fetchInterestPPCEffectiveDate	<p>&lt;p&gt;This method will check if interest ppc for the input product group code exists. If it does, this method will return the effective date for the interest ppc, else it will return null. &lt;/p&gt;           @param productGroupCode @param</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			domainCategory @return	
IPMUPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUPricingAdapter	fetchInterestPPCEffectiveDate	<p>This method will check if interest ppc for the input product group code exists. If it does, this method will return the effective date for the interest ppc, else it will return null. </p> @param productGroupCode @param domainCategory @return	<p>This method will check if the input offer is validated. If not, it will check if offer Start Date is greater than or equal to the PPC Effective Date for the Product Group of the offer. If yes, this method will mark the offer as validated and exit, else throw business exception. </p> @param offer
IPMUPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUPricingAdapter	fetchInterestPPCEffectiveDate	<p>This method will check if interest ppc for the input product group code exists. If it does, this method will return the effective date for the interest ppc, else it will return null. </p> @param productGroupCode @param domainCategory @return	<p>This method will do the following for the input product group code: <ol> <li>Validate Price Policy Chart for Interest</li> If fail, add exception - Price Policy Chart (Interest) not defined for the selected Product Group.   <li>Check if fee ppc exists and if it does perform the below, else update product group as validated. <ol> <li>Validate Product Group Role Mapping has been performed for the selected Product Group and PPC (Fees)</li> If fail, add exception - Product Group Role Mapping not performed for the selected Product Group.   </ol> <li>Validate For Product Group which is Asset, validate whether a PPC (Regular Interest) is defined for the ΓÇÿDomain CategoryΓÇÖ as ΓÇÿOriginationΓÇ?</li> If fail, add exception - Price Policy Chart (Regular Interest) not defined for Origination domain category for the selected Product Group. </ol> </p> @param productGroup

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUPricingAdapter	fetchFeePPCDetails	<p>This method will check if fee ppc for the input product group code exists. </p> @param productGroupCode @param domainCategory @return	
IPMUPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUPricingAdapter	fetchFeePPCDetails	<p>This method will check if fee ppc for the input product group code exists. </p> @param productGroupCode @param domainCategory @return	<p>This method will do the following for the input product group code: <ol> <li>Validate Price Policy Chart for Interest</li> If fail, add exception - Price Policy Chart (Interest) not defined for the selected Product Group.   <li>Check if fee ppc exists and if it does perform the below, else update product group as validated. <ol> <li>Validate Product Group Role Mapping has been performed for the selected Product Group and PPC (Fees)</li> If fail, add exception - Product Group Role Mapping not performed for the selected Product Group.   </ol> <li>Validate For Product Group which is Asset, validate whether a PPC (Regular Interest) is defined for the ΓÇÿDomain CategoryΓÇÖ as ΓÇ£OriginationΓÇ?</li> If fail, add exception - Price Policy Chart (Regular Interest) not defined for Origination domain category for the selected Product Group. </ol> </p> @param productGroup
ITwoFactorAuthServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITwoFactorAuthServiceAdapter	sendSMS		
ITwoFactorAuthServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITwoFactorAuthServiceAdapter	sendSMS		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITwoFactorAuthServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ITwoFactorAuthServiceAdapter	resendSMS		
ITwoFactorAuthServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ITwoFactorAuthServiceAdapter	validate		
IStatementNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IStatementNotificationAdapter	raiseCreditAlert		
IStatementNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\alert\IStatementNotificationAdapter	raiseCreditAlert		Financial statement item will be logged for demand deposit account on stimulus to financial event which generally changes {@link DemandDepositBalances}.like cash deposit,cash withdrawal ,fund transfer,cheque deposit etc. This method is used to create new statement item. @param ddFinStmntItem,financial statement item for demand deposit account. @param ddFinStmntItem

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalWorkListAdapter	batchAuthorize	@param sessionContext @param batchId @param batchTyp @return @throws FatalException	This method Authorizes the underlying entries for a batch code  And updates <b><i>[entries_status_flag]</b></i> in <i>[FLX_JE_BATCH_CODES_ALL_B]</i> table  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,JournalEntryBatchAppDTO,journalEntryBatchAppDTO, ,journalEntryBatchAppDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IJournalWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalWorkListAdapter	batchDelete	@param sessionContext @param batchId @param taskCode @return @throws FatalException	This method Deletes the underlying entries for a batch code  And updates <b><i>[entries_status_flag]</b></i> in <i>[FLX_JE_BATCH_CODES_ALL_B]</i> table  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,JournalEntryBatchAppDTO,journalEntryBatchAppDTO, ,journalEntryBatchAppDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBeneficiaryDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IBeneficiaryDetailsAdapter	getBeneficiaries	@return	This method returns all beneficiaries account numbers, to whom money are transferred. @return List,beneficiaries, , List containing beneficiary account numbers.
IBeneficiaryDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IBeneficiaryDetailsAdapter	getAccountId	@return	This method returns all beneficiaries account numbers, to whom money are transferred. @return List,beneficiaries, , List containing beneficiary account numbers.
IBeneficiaryDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IBeneficiaryDetailsAdapter	getAccountId	@return	This method returns the account-id from which money are getting transferred. @return String,accountId, , Indicates the account Id.
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	publishRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	publishRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	validateRuleId		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	viewRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	viewRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	modifyRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	modifyRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This service is used for resolving the template for financial profile capture. An applicant can apply to multiple product groups within the same submission.   Based on the information provided for the applicant , the product group one applies to and the purpose for application different templates may be resolved.  This information is provided to the rule engine which returns a template identifier per product group.  If an applicant applies to multiple product groups, the priorities for all templates are compared and the one  with the highest priority is stamped against the applicant within the submission. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param sessionContext @param submissionInputDTO @return @throws FatalException
IOrigination RulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		
IOrigination RulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This service will execute the rule for Fast Refinance if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for fast refinance rule. @return Boolean, Returns if fast refinance is applicable or not @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This service will execute the rule for Employee benefits if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for Employee benefits rule. @return Boolean, Returns if Employee benefits are applicable or not @throws FatalException
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This service will execute the rule for Low Doc benefits if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for Low Doc rule. @return Boolean, Returns if Low Doc benefits are applicable or not @throws FatalException
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		<p>The method is called from the Funding Table in UI to inquire the LMI quote from the LMI provider.&lt;br&gt;The applicant details and collateral details are required in addition to LendingProductGroupLinkageDTO&lt;br&gt;To get the Quote. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO , This stores information &lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @param ApplicantDataDTO [] applicantDataDTOs , This contains the details of the applicants who &lt;br&gt; have applied for the product group. @param CollateralDTO [] collateralDTOs , This contains the collateral details linked to the product group. @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001 &lt;br&gt; This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				thrown when Rule Id is not linked in Origination Preference  for LRM Submission Type and LRM Provider Rules.
IOrigination RulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		<p>This method needs to be called if LRM is applicable at the product group level and will do the following:&lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;1.calculate LVR, execute rule to get submission type and provider type&lt;/li&gt;</li> <li>&lt;li&gt;2.Execute rule to get if DUA is applicable&lt;/li&gt;</li> </ul> <p>@param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information &lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001 &lt;br&gt; This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.OR_LRM_NOT_APPLICABLE : FC_OR_MI_0001 &lt;br&gt; This exception is thrown when we try to execute LRM rules &lt;br&gt; for a Product Group for which LMI is not applicable. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This method will execute Rules for LRM policy and update the {@link LendingProductGroupLinkage} with the results. It will also return the policy execution results to the caller. @param sessionContext the {@link SessionContext} @param productGroupLinkageDTO the {@link LendingProductGroupLinkageDTO} @return the {@link LRMApplicableResponse} @throws FatalException
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType   LRM required is obtained from Origination Preference. Lending Product Group Linkage related facts are   sent to the rule engine to execute rule for that rule Id whose response will be LRM Required or not. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType  LRM Submission is obtained from Origination Preference.LVR value and Lending   Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMSubmissionType
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType  LRM Submission and LRM Provider is obtained from Origination Preference.LVR value and Lending   Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   The response is populated in LRM rule Execution Response.  @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMRuleExecutionResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRule		<p>This service is used to validate the applicants applying for different variants under a submission.&lt;br&gt; These options are made available based on rules setup within the system. These options are granted only&lt;br&gt; when the rules return a positive outcome.&lt;br&gt; Existing applicants that are employees can apply for employee benefits.&lt;br&gt; For Refinancing of loans, fast refinance option is available.&lt;br&gt; For applicants with employment restrictions, option for low doc is available.&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>indicating if the invocation succeeded or failed.  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#APPLICANT_SUBMISSION_VALIDATION_FAILED} : FC_OR_DE_0088 &lt;br&gt;  This exception occurs applicants applying for variants are not applicable. Business validation error is thrown and variants are not selected. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#EMPLOYEE_BENEFITS_NOT_APPLICABLE} : FC_OR_DE_0080 &lt;br&gt; This exception occurs when employee benefits flag is checked for a Product Group &lt;br&gt; but Employee benefit Applicability Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#LOW_DOC_NOT_APPLICABLE} : FC_OR_DE_0079 &lt;br&gt; This exception occurs when Low Doc Applicable flag is checked for a Product Group &lt;br&gt; but Low doc Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_AUTO_REFI_NOT_APPLICABLE} : FC_OR_AR_0001 &lt;br&gt; This exception occurs when Fast Refi flag is checked for a Product Group &lt;br&gt; but Fast Refi Rule Execution returns Not Applicable.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRuleForFastRefinance	This service will execute the rule for Fast Refinance if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO,, Indicates the submission details being validated for fast refinance rule. @return Boolean, Returns if fast refinance is applicable or not @throws FatalException	This service is used to validate the applicants applying for different variants under a submission.  These options are made available based on rules setup within the system. These options are granted only  when the rules return a positive outcome.  Existing applicants that are employees can apply for employee benefits.  For Refinancing of loans, fast refinance option is available.  For applicants with employment restrictions, option for low doc is available.  @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception   {@link OriginationErrorConstants#APPLICANT_SUBMISSION_VALIDATION_FAILED} : FC_OR_DE_0088   This exception occurs applicants applying for variants are not applicable. Business validation error is thrown

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>and variants are not selected. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#EMPLOYEE_BENEFITS_NOT_APPLICABLE} : FC_OR_DE_0080 &lt;br&gt; This exception occurs when employee benefits flag is checked for a Product Group &lt;br&gt; but Employee benefit Applicability Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#LOW_DOC_NOT_APPLICABLE} : FC_OR_DE_0079 &lt;br&gt; This exception occurs when Low Doc Applicable flag is checked for a Product Group &lt;br&gt; but Low doc Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_AUTO_REFI_NOT_APPLICABLE} : FC_OR_AR_0001 &lt;br&gt; This exception occurs when Fast Refi flag is checked for a Product Group &lt;br&gt; but Fast Refi Rule Execution returns Not Applicable.&lt;br&gt;</p>
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRuleForEmployeeBenefits	<p>This service will execute the rule for Employee benefits if applicable.  @fcb.param  MI,SessionContext,sessionContext,  ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param  MI,SubmissionInputDTO,  submissionInputDTO, , Indicates the submission details being validated for Employee benefits rule. @return Boolean, Returns if Employee benefits are applicable or not  @throws FatalException</p>	<p>This service is used to validate the applicants applying for different variants under a submission.&lt;br&gt; These options are made available based on rules setup within the system. These options are granted only&lt;br&gt; when the rules return a positive outcome.&lt;br&gt; Existing applicants that are employees can apply for employee benefits.&lt;br&gt; For Refinancing of loans, fast refinance option is available.&lt;br&gt; For applicants with employment restrictions, option for low doc is available.&lt;br&gt; @param  SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @param SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#APPLICANT_SUBMISSION_VALIDATION_FAILED} : FC_OR_DE_0088 &lt;br&gt; This exception occurs applicants applying for variants are not applicable. Business validation error is thrown and variants are not selected. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#EMPLOYEE_BENEFITS_NOT_APPLICABLE} : FC_OR_DE_0080 &lt;br&gt; This exception occurs when employee benefits flag is checked for a Product Group &lt;br&gt; but Employee benefit Applicability Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#LOW_DOC_NOT_APPLICABLE} : FC_OR_DE_0079 &lt;br&gt; This exception occurs when Low Doc Applicable flag is checked for a Product Group &lt;br&gt; but Low doc Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_AUTO_REFI_NOT_APPLICABLE} : FC_OR_AR_0001 &lt;br&gt; This exception occurs when Fast Refi flag is checked for a Product Group &lt;br&gt; but Fast Refi Rule Execution returns Not Applicable.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeRuleForLowDoc	<p>This service will execute the rule for Low Doc benefits if applicable.</p> <p>@fcb.param  MI,SessionContext,sessionContext,  ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param  MI,SubmissionInputDTO,  submissionInputDTO,, Indicates the submission details being validated for Low Doc rule. @return Boolean, Returns if Low Doc benefits are applicable or not @throws FatalException</p>	<p>This service is used to validate the applicants applying for different variants under a submission.&lt;br&gt; These options are made available based on rules setup within the system. These options are granted only&lt;br&gt; when the rules return a positive outcome.&lt;br&gt; Existing applicants that are employees can apply for employee benefits.&lt;br&gt; For Refinancing of loans, fast refinance option is available.&lt;br&gt; For applicants with employment restrictions, option for low doc is available.&lt;br&gt; @param  SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param SubmissionInputDTO, submissionInputDTO, ,This class holds the entire information of one submission and will be populated via different channels BANKER, PORTAL etc It consists of SubmissionHeader {@link com.ofss.fc.app.origination.dto.core.submission.HeaderDTO} an array of ApplicantData {@link com.ofss.fc.app.origination.dto.core.application.ApplicantDataDTO} and an array of ProductDetails {@link com.ofss.fc.app.origination.dto.core.application.ProductDetailsDTO} . Refer to {@link com.ofss.fc.app.origination.dto.core.submission.SubmissionInputDTO} for more details. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#APPLICANT_SUBMISSION_VALIDATION_FAILED} : FC_OR_DE_0088 &lt;br&gt; This exception occurs applicants applying for variants are not applicable. Business validation error is thrown</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>and variants are not selected. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#EMPLOYEE_BENEFITS_NOT_APPLICABLE} : FC_OR_DE_0080 &lt;br&gt; This exception occurs when employee benefits flag is checked for a Product Group &lt;br&gt; but Employee benefit Applicability Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#LOW_DOC_NOT_APPLICABLE} : FC_OR_DE_0079 &lt;br&gt; This exception occurs when Low Doc Applicable flag is checked for a Product Group &lt;br&gt; but Low doc Rule Execution returns Not Applicable.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_AUTO_REFI_NOT_APPLICABLE} : FC_OR_AR_0001 &lt;br&gt; This exception occurs when Fast Refi flag is checked for a Product Group &lt;br&gt; but Fast Refi Rule Execution returns Not Applicable.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	populateLowDocFacts	<p>This service will populate the fact required for the Low Doc rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details @fcb.param MI, Map&lt;String, CollateralDTO&gt;, collateralMap, , Indicates the collateral details with collateral code as the key. Collaterals linked to the facility are fetched using this collateral code. @fcb.param MI,Map&lt;Integer, ApplicantDataDTO&gt;, applicantMap, , Indicates the map containing applicant data with applicant serial number as the key. @return FactInformationDTO[], Returns a list of facts required for execution of Low Doc Rule. @throws FatalException</p>	<p>This service will execute the rule for Low Doc benefits if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for Low Doc rule. @return Boolean, Returns if Low Doc benefits are applicable or not @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	populateEmployeeFacts	<p>This service will populate the fact required for the Employee rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant for the product group @fcb.param MI,ApplicantRelationshipDTO[], applicantRelationshipDTO, , Indicates the information about relationship between applicants of a submission. @fcb.param MI,Map&lt;Integer, ApplicantDataDTO&gt;applicantMap, , Indicates map of applicant data with serial number as the key. @return FactInformationDTO[], List of facts to be passed for Employee Rule Execution @throws FatalException</p>	<p>This service will execute the rule for Employee benefits if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for Employee benefits rule. @return Boolean, Returns if Employee benefits are applicable or not @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	populateFactsForFastRefinance	<p>This service will populate the fact required for the Fast Refi Rule to be executed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant for the product group @fcb.param MI,LendingProductGroupLinkageDTO, lendingProductGroupLinkageDTO, , Indicates the lending product group details @fcb.param MI,CollateralDTO, collateralDTO, , Indicates the collateral data @fcb.param MI, ApplicationChannelType applicationChannelType, , Indicates the application channel @return FactInformationDTO[], List of facts to be passed for Fast Refi Rule Execution @throws FatalException</p>	<p>This service will execute the rule for Fast Refinance if applicable. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,SubmissionInputDTO, submissionInputDTO, , Indicates the submission details being validated for fast refinance rule. @return Boolean, Returns if fast refinance is applicable or not @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	fetchRuleNameByRuleId		<p>This service will return the Origination preferences maintained in the system for the market entity and business unit combination&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,OriginationPreferenceDTO ,originationPreferenceDTO, {@link com.ofss.fc.app.origination.dto.core.preference.OriginationPreferenceDTO} This object holds the Origination preferences @return {@link com.ofss.fc.app.origination.dto.core.preference.OriginationPreferenceInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#ORIG_PREF_NOT_MAINTAINED} : FC_OR_00012&lt;br&gt; This exception is thrown when Origination Preferences are not maintained&lt;br&gt; for the logged in user's Business Unit and Market Entity.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeLRMRequired		<p>The method is called from the Funding Table in UI to inquire the LMI quote from the LMI provider.&lt;br&gt;The applicant details and collateral details are required in addition to LendingProductGroupLinkageDTO&lt;br&gt;To get the Quote. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param LendingProductGroupLinkageDTO productGroupLinkageDTO , This stores information &lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @param ApplicantDataDTO [] applicantDataDTOs , This contains the details of the applicants who &lt;br&gt; have applied for the product group. @param CollateralDTO [] collateralDTOs , This contains the collateral details linked to the product group. @return TransactionStatus @exception OriginationErrorConstants.OR_RULE_EXEC_FAILED : FC_OR_MI_0001 &lt;br&gt; This exception is thrown when any of the LRM rule execution fails. @exception OriginationErrorConstants.RULE_ID_NOT_MAINTAINED : FC_OR_00011 &lt;br&gt; This exception is thrown when Rule Id is not linked in Origination Preference&lt;br&gt; for LRM Submission Type and LRM Provider Rules.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeLRMRequired		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType  LRM required is obtained from Origination Preference. Lending Product Group Linkage related facts are   sent to the rule engine to execute rule for that rule Id whose response will be LRM Required or not. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeLRMRequired		This method needs to be called if LRM is applicable at the product group level and will do the following: <li>1.Execute rule to check if LRM is required</li><li>2. If LRM is required,calculate LVR, execute rule to get submission type and provider type</li>@fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,CreditFacilityDTO[] facilityDetailsDTOs , This contains the various collaterals linked to the Product Group. @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeDUAApplibilityRule		This method executes rules to determine if DUA is applicable for LRM. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeSubmissionTypeRule		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType   LRM Submission is obtained from Origination Preference.LVR value and Lending   Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMSubmissionType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationRulesAdapter	executeLRMProviderRule		This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType   LRM Submission and LRM Provider is obtained from Origination Preference.LVR value and Lending   Product Group Linkage related facts are sent to the rule engine to execute rule for that rule Id.   The response is populated in LRM rule Execution Response.  @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information   regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @fcb.param MI,BigDecimal lvrValue , The loan to value ratio @return LRMRuleExecutionResponse
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	This method delegates the call to maintainWorkItem method in WorkListApplication. @param commitConditions @param isTransactionFailue @param txnKey @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	This method will fetch the required authorizations raised for the undergoing transaction from  the FCRTThreadAttribute and derive the corresponding Executable WorkFlow.   For a single transaction various authorizations can be raised and in such cases multiple   Executable WorkFlows will be generated and the corresponding reasons/WorkFlow types will also   be returned to the calling function.  @param requiredAuthorizationMap

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@return Map<Object, Object>
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	This method is the entry point for workList processing. This method takes care of   creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	Every workItem created will have certain number of approvals needed before it is finally approved when final approval has been done then the workItem status is marked as APPROVED and it's  processing status is changed to COMPLETED. @param sessionContext @param processStatus @param workItemDTO @param status @param workItem @param inputParameterMap @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	This method is the entry point for workList processing. This method takes care of  creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	maintainWorkItem	This method delegates the call to maintainWorkItem method in WorkListApplication. @param sessionContext @param commitConditions @param isTransactionFailue @param externalRefNumber @param txnKey	Every workItem created will have certain number of approvals needed before it is finally approved  when final approval has been done then the workItem status is marked as APPROVED and it's   processing status is changed to COMPLETED.  @param sessionContext @param processStatus @param workItemDTO @param status @param workItem @param inputParameterMap @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteFromWorkItem	This method delegates the call to deleteFromWorkItem method in WorkListApplication. @param userId @param keyAsString @param taskCode	This method delegates the call to deleteFromWorkItem method in WorkListApplication. @param userId @param keyAsString @param taskCode
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteFromWorkItem	This method delegates the call to deleteFromWorkItem method in WorkListApplication. @param userId @param keyAsString @param taskCode	This service provides functionality to delete a workItem for the given keyAsString, taskCode combination.  When a workItem is deleted all the underlying ExecutableWorkflow, ExecutableWorkFlowStage and the   will also get deleted, any approvals which have been received for the workItem will also be lost,   but the history of actions performed on the workItem will be maintained.  @param sessionContext @param keyAsString @param taskCode @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteFromWorkItem	This method delegates the call to deleteFromWorkItem method in WorkListApplication. @param userId @param keyAsString @param taskCode	This service provides functionality to delete a workItem for the given keyAsString, taskCode combination.  When a workItem is deleted all the underlying ExecutableWorkflow, ExecutableWorkFlowStage and the   will also get deleted, any approvals which have been received for the workItem will also be lost,   but the history of actions performed on the workItem will be maintained.  @param sessionContext @param keyAsString @param taskCode @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteFromWorkItem	This method delegates the call to deleteFromWorkItem method in WorkListApplication. @param userId @param keyAsString @param taskCode	Function to cancel maintenance operations which have not yet been approved.   It checks if the domain object to be removed exists in the database. If not found then it will throw  BusinessException with the following error Code ErrorConstants#MID_EREC_NOT_FOUND  If a record is found it checks whether the creator of record is same as current user  if they are not same then BusinessException with error code   FCWorkListErrorConstants.ONLY_INITIATOR_CAN_PERFORM_THIS_ACTION is thrown .  If the record has been already approved then BusinessException with error code  FCWorkListErrorConstants.APPROVAL_STAGE_IS_ALREADY_COMPLETE is thrown.  <b>Note :- This function will not delete records from base tables. Records in base tables  can only be closed (close function is being provided for the purpose).</b>  @param abstractDomainObject @param abstractRepository

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	This method fetches WorkItem for the provided keyAsString value and then finds whether the operation represented by the provided maintenanceType can be performed on the current transaction or not. @param sessionContext @param keyAsString @param taskCode @param maintenanceType @param userId @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	This method fetches WorkItem for the provided keyAsString value and then finds whether the operation represented by the provided maintenanceType can be performed on the current transaction or not. @param sessionContext @param keyAsString @param taskCode @param maintenanceType @param userId @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param	Function to create a new instance of the domain Entity and persist It  It checks if any object with the same key already exists or has been "closed"  If yes then function will throw an Business Exception with the following Error Code ErrorConstants#MID_MSG_DBDUPLROW  @param abstractDomain @param abstractRepository

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			userId	
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	Function to update an existing Domain Object As part of updation the function will fetch the existing object  If not found then it will throw BusinessException with the following error Code ErrorConstants#MID_EREC_NOT_FOUND Version Number is used here for optimistic Concurrency Control  If the version no passed in the input is not equal to -1 then function will check the input version no and the existing version no  If found different then it will throw an Business Exception with the Error Code ErrorConstants#MID_INVALID_RECORD_VERSION  If the passed version is equal to -1 then version validations will be bypassed and the record will be updated blindly  @param abstractDomainObject @param abstackRepository
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	Function to merge(Updates) the Domain object with the new values as merge operation is more intelligent than update as no need to fetch the existing object from the database, the newly created domain object overwrites its property to existing object, Provided hibernate VERSION must be specified in .hbm file instead of version as property. ErrorConstants#MID_EREC_NOT_FOUND Version Number is used here for optimistic Concurrency Control  If object found with stale state then it hibernate will throw an Business Exception with the Error Code ErrorConstants#MID_INVALID_RECORD_VERSION  @param abstractDomainObject @param abstackRepository

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	Function to close an existing domain object  As part of the closure function will fetch the existing object  If not found then it will throw BusinessException with the following error Code ErrorConstants#MID_EREC_NOT_FOUND  Version Number is used here for optimistic Concurrency Control  If the version no passed in the input is not equal to -1 then function will check the input version no and the existing version no  If found different then it will throw an BusinessException with the Error Code ErrorConstants#MID_INVALID_RECORD_VERSION  If the passed version is equal to -1 then version validations will be bypassed and the record will be updated blindly  A closed Domain object will not be fetched in normal Inquiries but only when closed records are fetched  @param abstractDomainObject @param abstractRepository
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	testActionPerformable	This method delegates the call to testActionPerformable which fetches the record from workitem and checks whether the operation represented by maintenanceType can be performed or not. @param keyAsString @param taskCode @param maintenanceType @param userId	Function to reopen a closed domain Entity The method will first fetch the existing record  Then status of existing object is checked and if the status is Active   an exception with the error code CommonErrorConstants#MID_ACTIVATE_ACTIVE_RECORD  is thrown. @param abstractDomainObject @param abstractRepository
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call  webServices to initiate common approval process.  @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method creates a binding stub and uses that stub to call  webServices to initiate common approval process.  @param taskCode @param workItemId @param initiatorId @param maintenanceType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method will trigger call for initiating Approval Process by delegating a call to IWorklistAdapter which in turn used a stub to call the remote webService and initiates the approval process. This method will be called when the underlying application service is being invoked from a BPEL process if the application service is not invoked by the BPEL process in that case the approvalProcess will be invoked after the WorkItem has been raised. @param SessionContext sessionContext @param String entityKey @param String taskCode @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for creating workItem and based on the response triggers a call to initiate common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for creating workItem and based on the response triggers a call to initiate common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call Initiating approval process after committing data in flx_wl_workitem and flx_wl_workitem_dtls table. If commit fails approval process will not be initiated. If approval process initiation fails, the committed row from both the tables will be deleted. @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for updating an existing WorkItem, at the time of update workItem status, process status and reference number will be updated and based on the response a call will be triggered for initiating common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call. If the record has been modified by the initiator in that no web service call will be triggered to initiate common approval process. @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method will trigger call for initiating Approval Process by delegating a call to IWorklistAdapter which in turn used a stub to call the remote webService and initiates the approval process. This method will be called when the underlying application service is being invoked from a BPEL process if the application service is not invoked by the BPEL process in that case the approvalProcess will be invoked after the WorkItem has been raised. @param SessionContext sessionContext @param String entityKey @param String taskCode @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for creating workItem and based on the response triggers a call to initiate common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for creating workItem and based on the response triggers a call to initiate common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	initiateCommonApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param executableWorkflowId @param initiatorId @param maintenanceType	This method calls domain level service for updating an existing WorkItem, at the time of update workItem status, process status and reference number will be updated and based on the response a call will be triggered for initiating common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process in BPEL. The initiation internally happens through a web service call. If the record has been modified by the initiator in that no web service call will be triggered to initiate common approval process. @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	callApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param workitemId @param approvalProcessKey @throws FatalException	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param workItemId @param initiatorId @param maintenanceType
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	callApprovalProcess	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param workitemId @param approvalProcessKey @throws FatalException	This method creates a binding stub and uses that stub to call webServices to initiate common approval process. @param taskCode @param workitemId @param approvalProcessKey @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteDraft	This method delegates call to deleteDraft method in DraftApplicationService which deletes draft data from repository. @param taskCode @param keyAsString @param	This method delegates call to deleteDraft method in DraftApplicationService which deletes draft data from repository. @param taskCode @param keyAsString @param userId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			userId	
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteDraft	This method delegates call to deleteDraft method in DraftApplicationService  which deletes draft data from repository.  @param taskCode @param keyAsString @param userId	This operation will add WorkItem into WorkList application. WorkItem can be accompanied by authorization reason codes for required authorizations and DefinedWorkflowDTOs of those authorizations. @param sessionContext SessionContext @param workItemDTO WorkItemDTO contains the details of WorkItem @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteDraft	This method delegates call to deleteDraft method in DraftApplicationService  which deletes draft data from repository.  @param taskCode @param keyAsString @param userId	This operation will add WorkItem into WorkList application. WorkItem can be accompanied by authorization reason codes for required authorizations and DefinedWorkflowDTOs of those authorizations. @param sessionContext SessionContext @param workItemDTO WorkItemDTO contains the details of WorkItem @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	deleteDraft	This method delegates call to deleteDraft method in DraftApplicationService  which deletes draft data from repository.  @param taskCode @param keyAsString @param userId	Function to create a new instance of the domain Entity and persist It  It checks if any object with the same key already exists or has been "closed"  If yes then function will throw an Business Exception with the following Error Code ErrorConstants#MID_MSG_DBDUPLROW  @param abstractDomain @param abstractRepository

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getExecutableWorkflows	This method will fetch the required authorizations raised for the undergoing transaction from  the FCRThreadAttribute and derive the corresponding Executable WorkFlow.   For a single transaction various authorizations can be raised and in such cases multiple   Executable WorkFlows will be generated and the corresponding reasons/WorkFlow types will also   be returned to the calling function.  @param requiredAuthorizationMap @return Map<Object, Object>	This operation will update an existing WorkItem with new details,authorization reasons and new DefinedWorkflows. Updated details of WorkItem will come in WorkItemDTO. Old ExecutableWorkflows will be discarded and updated with new DefinedWorkflows. @param sessionContext @param workItemDTO @param workflowReasons @param definedWorkflows @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getExecutableWorkflows	This method will fetch the required authorizations raised for the undergoing transaction from  the FCRThreadAttribute and derive the corresponding Executable WorkFlow.   For a single transaction various authorizations can be raised and in such cases multiple   Executable WorkFlows will be generated and the corresponding reasons/WorkFlow types will also   be returned to the calling function.  @param requiredAuthorizationMap @return Map<Object, Object>	This service will be used to process the workItems when at the time of final approval   some new workFlows are being raised which were not raised earlier.  The workFlows which were raised earlier would have been already processed and the  entries for these workFlows along with their stages will be deleted, but a history of   all the operations performed on the workItem will be maintained.  All the workFlows which have been raised earlier and already processed will not be considered  for approval again, only the workFlows which have been not raised previously will be considered  for processing.  The new workFlows raised will be processed against the workItem id which was generated earlier  and the transaction will get approved ones all the newly raised workFlows have been approved.  @param sessionContext @param workItemDTO @return WorkItemResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getExecutableWorkflows	This method will fetch the required authorizations raised for the undergoing transaction from the FCRThreadAttribute and derive the corresponding Executable WorkFlow.   For a single transaction various authorizations can be raised and in such cases multiple   Executable WorkFlows will be generated and the corresponding reasons/WorkFlow types will also   be returned to the calling function.  @param requiredAuthorizationMap @return Map<Object, Object>	This operation will update an existing WorkItem with new details,authorization reasons and new DefinedWorkflows. Updated details of WorkItem will come in WorkItemDTO. Old ExecutableWorkflows will be discarded and updated with new DefinedWorkflows. @param sessionContext @param workItemDTO @param workflowReasons @param definedWorkflows @return TransactionStatus @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getExecutableWorkflows	This method will fetch the required authorizations raised for the undergoing transaction from the FCRThreadAttribute and derive the corresponding Executable WorkFlow.   For a single transaction various authorizations can be raised and in such cases multiple   Executable WorkFlows will be generated and the corresponding reasons/WorkFlow types will also   be returned to the calling function.  @param requiredAuthorizationMap @return Map<Object, Object>	This service will be used to process the workItems when at the time of final approval   some new workFlows are being raised which were not raised earlier.  The workFlows which were raised earlier would have been already processed and the  entries for these workFlows along with their stages will be deleted, but a history of   all the operations performed on the workItem will be maintained.  All the workFlows which have been raised earlier and already processed will not be considered  for approval again, only the workFlows which have been not raised previously will be considered  for processing.  The new workFlows raised will be processed against the workItem id which was generated earlier  and the transaction will get approved ones all the newly raised workFlows have been approved.  @param sessionContext @param workItemDTO @return WorkItemResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getValidAuthorizationsForCurrentTransaction	<p>This method fetches all the received and pending authorizations for the passed in WorkItemId&lt;br&gt;workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and &lt;br&gt; ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization &lt;br&gt; map since we don't need to raise fresh approval WorkFlows for authorizations which have been &lt;br&gt; already raised/approved for the current workItemVersion and maintenanceType.&lt;br&gt; @param workItemId @param workItemVersion @param MaintenanceType @return List&lt;String&gt; List of workflowReasons. @throws FatalException</p>	<p>This method will remove duplicate workFlows which have been raised on the current transaction. It may happen that one workFlow is being configured against more than one authorization configuration, in that case we are not required to raise two separate workFlows since in that case the same users would be approving the same data twice. The required authorizations would be fetched from FCRTThreadAttribute, and in case of duplicate workFlows the first Autorization Reason Code would be fetched while the later one would be discarded. This method would return list of all the valid workflowReasons. @return List&lt;String&gt; List of workflowReasons @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getValidAuthorizationsForCurrentTransaction	This method fetches all the received and pending authorizations for the passed in WorkItemId workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and   ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization   map since we don't need to raise fresh approval WorkFlows for authorizations which have been   already raised/approved for the current workItemVersion and maintenanceType.  @param workItemId @param workItemVersion @param MaintenanceType @return List<String> List of workFlowReasons. @throws FatalException	This method fetches all the received and pending authorizations for the passed in WorkItemId workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and   ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization   map since we don't need to raise fresh approval WorkFlows for authorizations which have been   already raised/approved for the current workItemVersion and maintenanceType.  @param requiredAuthorizationMap @param workItemId @param workItemVersion @param MaintenanceType @return List<String> List of workFlowReasons.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getValidAuthorizationsForCurrentTransaction	<p>This method fetches all the received and pending authorizations for the passed in WorkItemId&lt;br&gt;workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and &lt;br&gt; ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization &lt;br&gt; map since we don't need to raise fresh approval WorkFlows for authorizations which have been &lt;br&gt; already raised/approved for the current workItemVersion and maintenanceType.&lt;br&gt; @param workItemId @param workItemVersion @param MaintenanceType @return List&lt;String&gt; List of workFlowReasons. @throws FatalException</p>	<p>This method is called to perform post Amend checks.&lt;br&gt; After the maintenance is performedThis method is called to check that the Authorization reasons are same after performing amend operation.&lt;br&gt; @param sessionContext @param taskCode @param keyAsString @param userId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getValidAuthorizationsForCurrentTransaction	This method fetches all the received and pending authorizations for the passed in WorkItemId workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and   ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization   map since we don't need to raise fresh approval WorkFlows for authorizations which have been   already raised/approved for the current workItemVersion and maintenanceType.  @param workItemId @param workItemVersion @param MaintenanceType @return List<String> List of workFlowReasons. @throws FatalException	This method is the entry point for workList processing. This method takes care of   creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRTThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getValidAuthorizationsForCurrentTransaction	This method fetches all the received and pending authorizations for the passed in WorkItemId workItemVersion and maintenanceType combination by doing a union on WorkItemHistory and   ExecutableWorkflow data, these authorizations are then removed from the requiredAuthorization   map since we don't need to raise fresh approval WorkFlows for authorizations which have been   already raised/approved for the current workItemVersion and maintenanceType.  @param workItemId @param workItemVersion @param MaintenanceType @return List<String> List of workFlowReasons. @throws FatalException	This method is the entry point for workList processing. This method takes care of   creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRTThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	getSessionContext	Returns the SessionContext for the current session from AdapterContextHelper. @return SessionContext	This method will retrieve the service inputs and AuditItem from FCRTThreadAttribute   and populates the maintenance related data in the DomainObjectDTO instance, puts the  serialized DomainObjectDTO instance in the AuditItem and persists the AuditItem instance.  Whenever a maintenance operation is performed and it gets approved/autoApproved  an audit data will be logged for the operation performed.  The audit record will contain following information of the entity being stored  1. EntityKey representing the keyAsString 2. TaskCode representing the taskCode 3. ChannelId representing the channel of transaction 4. Version representing the version number 5. ClassName represents the className 6. CreatedBy represents the userName of the initiator 7. PostingDate 8. auditDate 9. operationType 10.serializedEntityDTO 11.serializedEntity In case of transaction where approval is required data will be moved to   Audit table on the final approval only, intermediate approvals will not   be persisting any data in audit.  @throws FatalException
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	amendWorkItem	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	amendWorkItem	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException	This method is the entry point for workList processing. This method takes care of  creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	amendWorkItem	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException	This method calls domain level service for updating an existing WorkItem, at the time of update  workItem status, process status and reference number will be updated and based on the response   a call will be triggered for initiating common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process   in BPEL. The initiation internally happens through a web service call.  If the record has been modified by the initiator in that no web service call will be triggered to  initiate common approval process. @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	amendWorkItem	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException	This method is the entry point for workList processing. This method takes care of  creating/updating an ApprovalWorkItem. If the approval being processed is the last  approval required then the approved record will be moved into the respective base  This method checks whether any authorizations are required for the current operation,  it does so by looking for data in Map provided by FCRTThreadAttribute.AUTHORIZATIONS_REQUIRED  if any data is found in the map it means the operation requires authorization,   the DefinedWorkFlow,RequiredAuthorization are picked from the Map and corresponding  DefinedWorkflowDTO instance is generated while an array of workflowReasons is populated   from the RequiredAuthorization.   If no authorizations are required then the method will not proceed further and returns  control back to the calling function ; if authorizations are required then a WorkItemDTO  instance is created and is populated using entity related information.  Based upon the Maintenance action an ApprovalWorkItem is created/updated/approved and   in case of create/update(not when updated by the initiator) an approval process is   initiated which in turn creates a workItem instance in the BPEL WorkList and assigns the  workItem to users having approval rights.  @param commitConditions @param isTransactionFailue @param status @param txnKey @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\worklist\dto\IWorklistAdapter	amendWorkItem	This method is called to update the workItem as a part of amend operation. This method will refresh the entity in the workItem with the modified entity. @param sessionContext @throws FatalException	This method calls domain level service for updating an existing WorkItem, at the time of update  workItem status, process status and reference number will be updated and based on the response  a call will be triggered for initiating common approval process. The initiation of common approval process is the point where a call will be made to the BPEL client to initiate a process  in BPEL. The initiation internally happens through a web service call. If the record has been modified by the initiator in that no web service call will be triggered to initiate common approval process. @param sessionContext @param workItemDTO @param status @param isModifiedByCreator @throws FatalException
IAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B]. @param accountingEventMap @return	This method is used to post Accounting JMS messages if any. @throws FatalException
IAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B]. @param accountingEventMap @return	
IAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B]. @param accountingEventMap @return	This method is used to post Accounting JMS messages if any @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B]. @param accountingEventMap @return	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B]. @param accountingEventList @return
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to update product ledger balances. @param accountingEventList @return	This method is used to post Accounting JMS messages if any. @throws FatalException
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to update product ledger balances. @param accountingEventList @return	
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to update product ledger balances. @param accountingEventList @return	This method is used to post Accounting JMS messages if any @throws FatalException
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueue	This method is used to put JMS messages into the queue for further processing to update product ledger balances. @param accountingEventList @return	This method is used to put JMS messages into the queue for further processing to update product ledger balances. @param accountingEntryList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueueInBatchMode	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B] in batch mode. All the {@link AccountingEventKeyDTO}s will be posted using a single queue connection and a single object message. @param accountingEventMap @return @throws FatalException	This method is used to post Accounting JMS messages if any. @throws FatalException
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEventMessagesIntoQueueInBatchMode	This method is used to put JMS messages into the queue for further processing to generate accounting entries[FLX_AS_ACCT_ENTRY_B] in batch mode. All the {@link AccountingEventKeyDTO}s will be posted using a single queue connection and a single object message. @param accountingEventMap @return @throws FatalException	This method is used to post Accounting JMS messages if any @throws FatalException
IAccounting Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueueInBatchMode	This method is used to put JMS messages into the queue for further processing to update product ledger balances in batch mode. All the {@link AccountingEntryKeyDTO}s will be posted using a single queue connection and a single object message. @param accountingEntryMap @return @throws FatalException	This method is used to post Accounting JMS messages if any. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingAdapter	postEntryMessagesIntoQueueInBatchMode	This method is used to put JMS messages into the queue for further processing to update product ledger balances in batch mode. All the {@link AccountingEntryKeyDTO}s will be posted using a single queue connection and a single object message. @param accountingEntryMap @return @throws FatalException	This method is used to post Accounting JMS messages if any @throws FatalException
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates CCIWelcomeLetterReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the CoverLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to CoverLetter to make the final CoverLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates CoverReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the HypothecationLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to Hypothecation Letter to make the final HypothecationLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates HypothecationReceivingParty attributes for the HypothecationLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the PersonalGuaranteeLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to PersonalGuaranteeLetter to make the final PersonalGuaranteeLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates PersonalGuaranteeReceivingParty attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence <P> 1. Takes Domain Object as an input along with SessionContext. <P> 2. Populates the MortgageLetter Node as the starting tag for the extract. <P> 3. Calls different modules to populate the requested attributes for the extract. <P> 4. Appends all the attribute to Mortgage Letter to make the final MortgageLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates MortgageReceivingParty attributes for the MortgageLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	<p>This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the OfferLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to OfferLetter to make the final OfferLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationUIServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationUIServiceAdapter	fetchRecordsForHQL	This method will fetch the result for a given queryId and function component. It will take binding parameters of that query in form of Map. It takes hashcode as Input and checks whether any change has occurred in DB if hashcode is equals pickList hashcode then it will return null. @param sessionContext, Session Context which must contain the session ticket and user id. @param queryId, Id given for the HQL query @param funcComponentID, Function Component ID of LOV @param taskCode @param bindParams, contains the binding parameters for the query @param hashcode @return LOVResponse @throws FatalException	This method populates OfferReceivingParty attributes for the offerLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	debit	This method debits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt;</li> </ul> <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	debit	<p>This method debits the customer account for given txn amount</p> <p>@param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	debit	This method debits the customer account for given txn amount @paramsessionContext @paramaccountId @paramtxnAmount @paramlegder @paramtxnRate @paramlocalTxnDate @parammessageType @parameventCode @paramnarration @paramcardNumber @return @throwsFatalException	This method applies Acquirer and Forwarder Fee charges to a CASA account for a transaction of Instrument.   After Debiting the Customer Account by Total Fee amount, Acquirer and Forwarder GL's are credited by Acquirer and Forwarder amount respectively. Acquirer and Forwarder Fee may or may not be present for the given transaction.  @fcb.param MI,SessionContext,sessionContext,, This parameter captures the basic information about the user of the system. @fcb.param MI,FinancialContext,txnValue,, Financial Context holds basic information about the transaction.  @fcb.param MI,String,accountId,1..16, The CASA account number, for which Cash deposit transaction has to be performed @fcb.param MI,double,transactionAmount,, The amount in terms of the transaction currency @fcb.param MI,int,mnemonic,, Mnemonic code of the transaction @fcb.param MI,int,processingCode,, The processing code under which this transaction will be   registered in the system @fcb.param MI,int,transactionType,, Can be one of the values from <ul> <li>{@link AbstractApplication#NORMAL_TRANSACTION}</li> <li>{@link AbstractApplication#ADVICE_TRANSACTION}</li> </ul> @fcb.param MI,TransactionStatus,status,, The status of the transaction which indicates success or failure @return BaseResponse @throwsFatalException @throwsOverridableException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	debit	This method debits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method is used to debit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for debiting @param txnAmount the transaction amount to be used for debiting @param currencyCode the currency code to be used @return TransactionStatus will return success/failure the details after debit of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	debit	This method debits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method is used to removeEarmarkAndDebit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after removeEarmarkAndDebit of account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This operation enables you to perform a earmark purchase against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS</li> </ul> <p>@throws FatalException Please refer to the documentation above for <i>Transaction failure</i> section</p>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param ledger @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method is used to credit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for credit @param txnAmount the transaction amount to be used for credit @param currencyCode the currency code to be used @return TransactionStatus will return success/failure the details after credit of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountId the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	credit	This method credits the customer account for given txn amount @param sessionContext @param accountId @param txnAmount @param legder @param txnRate @param localTxnDate @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method is used to removeEarmarkAndCredit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after removeEarmarkAndDebit of account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark force purchase against a ATM Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	<p>This operation enables you to perform a earmark force purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method fetches the description for a particular hold defination @param holdDefintionId @return @throws FatalException
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method is used to earmark the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method is used to removeEarmarkAndDebit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after removeEarmarkAndDebit of account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method is used to removeEarmarkAndCredit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after removeEarmarkAndDebit of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	earmark	This method would add a earmark to a customer account @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method fetches the description for a particular hold defination @param sessionContext the session context to be used in fetching @param earmarkParametersId the earmark paramater id to be used for fetching the description @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	<p>This operation enables you to perform a earmark force purchase against a ATM Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	<p>This operation enables you to perform a earmark force purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	deleteEarmark	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	creditEarmark	This method creates a credit earmark which has been added @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt;</p> <p>&lt;b&gt;Transaction failure&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; Error Reason may be one of the following</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#ERR_INSUFF_FUNDS}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_ACCT_CODE}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#RECOR D_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSB K_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B LOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODR_ ONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_M ANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	creditEarmark	This method creates a credit earmark which has been added @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This operation enables you to perform a earmark purchase against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	creditEarmark	<p>This method creates a credit earmark which has been added</p> <p>@param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	creditEarmark	This method creates a credit earmark which has been added @param sessionContext @param accountId @param txnAmount @param localDate @param authCode @param description @param holdDefintionId @return @throws FatalException	This method deletes the earmark which was put previously @param sessionContext the session context to be used in interacting with the service @param txnAmount the transaction amount to be used for earmark @param accountID the account id to be used for earmark @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after earmark of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndDebit	Method which would be used for removal of earmark and debit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_F

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>UND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#RECOR D_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSB K_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_B LOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODR_ ONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_M ANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt;</li> </ul> <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndDebit	Method which would be used for removal of earmark and debit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndDebit	Method which would be used for removal of earmark and debit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndDebit	Method which would be used for removal of earmark and debit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @return @throws FatalException	This method is used to removeEarmarkAndDebit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after removeEarmarkAndDebit of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndCredit	Method which would be used for removal of earmark and credit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRE_01}  Transaction requires

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>authorisation for one of the reasons mentioned below</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I_B_WDR_LIM}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt;</p> <p>&lt;b&gt;Transaction failure&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIDD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOIDD}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; Error Reason may be one of the following</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYNC_ERROR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FATAL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS}</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>UFF_FUNDS}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#INSUFF_  FUNDS_OVL}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#OVER_I  B_WDR_LIM}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#INV_AC  CT_CODE}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#OLTP_I  NT_ERROR}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#ACCT_I  NFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#RECOR  D_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#INV_PSB  K_BAL}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#DB_ERR  OR}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#ACCT_C  LOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#ACCT_C  LOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#ACCT_B  LOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#NODR_  ONACCT}&lt;/li&gt; &lt;li&gt; {@link  com.ofss.fc.enumeration.ReplyMessageType#TOO_M  ANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt;  @throws FatalException Please refer to the  documentation above for &lt;i&gt;Transaction failure&lt;/i&gt;  section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndCredit	Method which would be used for removal of earmark and credit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	removeEarmarkAndCredit	Method which would be used for removal of earmark and credit the customer account @param sessionContext @param accountId @param txnAmount @param holdDefintionId @param ledger @param txnRate @param localTxnDate @param authCode @param description @param messageType @param eventCode @param narration @param cardNumber @return @throws FatalException	This method is used to removeEarmarkAndCredit the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for removeEarmarkAndDebit @param txnAmount the transaction amount to be used for removeEarmarkAndDebit @param currencyCode the currency code to be used @return ddaBalancedTO will fetch the details after removeEarmarkAndDebit of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireAccountBalance	This method is used to inquire balances for a given account number @param sessionContext @param accountId @return @throws FatalException	This operation gives the details of the CASA account balances such  as the available balance, the uncleared balance and the earmarked balance   using Balance Enquiry  These details are returned as a data transfer object of type DDABalancedTO  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate. etc&lt;br&gt; @return ATMBalanceResponse A response object which holds the details of the&lt;br&gt; balances for the given account.The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS} </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireAccountBalance	This method is used to inquire balances for a given account number @param sessionContext @param accountId @return @throws FatalException	<p>This operation enables you to perform an earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireAccountBalance	This method is used to inquire balances for a given account number @param sessionContext @param accountId @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireAccountBalance	This method is used to inquire balances for a given account number @param sessionContext @param accountId @return @throws FatalException	
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireAccountBalance	This method is used to inquire balances for a given account number @param sessionContext @param accountId @return @throws FatalException	This method is used to inquire the customer account balances by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for inquire @param txnAmount the transaction amount to be used for inquire @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after inquire of account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	checkAccountDetails	This method is used to check if the account number is valid @param sessionContext @param accountId @return @throws FatalException	<p>This operation allows for cheque book request for number of cheque leaves to be issued, no of cheque book to be issued &lt;br&gt; &lt;br&gt; These details are returned as a data transfer object of type ATMBalanceResponse&lt;br&gt; @fcb.param MI, ATMChannelContext, channelContext, , ATM Channel Context which must contain the ATM Key.&lt;br&gt; @fcb.param MI, ISOBusinessContext, data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIDD}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt;</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ErrorConstants#MID_EUT_INVALID_USERID}</li> <li>{@link ErrorConstants#MID_EST_INV_BANK}</li> </ul> &@throws FatalException Please refer to the documentation above for <i>Transaction failure</i> section
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	checkAccountDetails	This method is used to check if the account number is valid @param sessionContext @param accountId @return @throws FatalException	
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	checkAccountDetails	This method is used to check if the account number is valid @param sessionContext @param accountId @return @throws FatalException	This method is used to checking the account details of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for validating @return TransactionStatus will success or failure of account
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireMiniStatement	This method does a mini statement inquiry for given account number @param sessionContext @param accountId @return @throws FatalException	This operation should be use to inquire upon the statement of  accounts for a CASA account. The operation fetches all the transactions   done on a particular account in the specified period. This includes both  customer-initiated transactions and system initiated transactions like  interest and SC capitalization.   The statement of accounts for the current period fetches the details of all  transactions conducted during the current statement period. Similarly the  previous statement period fetches all the transactions performed during  the previous statement period.  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>payment transaction.&lt;br&gt; @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.&lt;br&gt; @return ATMStatementResponse The response contains details of the all transactions as well as&lt;br&gt; the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; following will be in the response&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_I NVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_BANK}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_C ORRDFN_NOT_FOUND}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_DUM MY_ERROR_CODE}&lt;/li&gt; &lt;/ul&gt; @throws FatalException See the return paragraph above</p>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireMini Statement	This method does a mini statement inquiry for given account number @param sessionContext @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	inquireMiniStatement	This method does a mini statement inquiry for given account number @param sessionContext @param accountId @return @throws FatalException	This method is used to inquireMiniStatement the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for inquireMiniStatement @return DDAMiniStatementDTO will fetch the details after inquireMiniStatement of account
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at a particular address,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param address @return @throws FatalException	This operation allows for cheque book request for number of cheque leaves to be issued,no of cheque book to be issued     These details are returned as a data transfer object of type ATMBalanceResponse  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>     <b>Transaction failure</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIID}</li> </ul> Error Reason may be one of the following <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}</li> <li>{@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt; {@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt; {@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt; {@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at a particular address,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param address @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at a particular address,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param address @return @throws FatalException	This method is used to issue cheque book of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for cheque book request @return TransactionStatus if cheque book request logging to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at a particular address,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param address @return @throws FatalException	This method is used to issue cheque book of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for cheque book request @return TransactionStatus if cheque book request logging to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at branch,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param	This operation allows for cheque book request for number of cheque leaves to be issued,no of cheque book to be issued     These details are returned as a data transfer object of type ATMBalanceResponse  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>isPersonalized @param requestDate  @param city @param branchId  @return @throws FatalException</p>	<p>transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;  &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIDD}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at branch,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param branchId @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at branch,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param branchId @return @throws FatalException	This method is used to issue cheque book of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for cheque book request @return TransactionStatus if cheque book request logging to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	issueChequeBook	This method does a cheque book request for a given account number at branch,it uses no of cheque books,no of cheque leaves,etc as input @param sessionContext @param accountId @param noOfChequeLeaves @param noOfChequeBooks @param isNicknameRequired @param isPersonalized @param requestDate @param city @param branchId @return @throws FatalException	This method is used to issue cheque book of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for cheque book request @return TransactionStatus if cheque book request logging to CASA failed or was success

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	transferFunds	This method is used for transferring funds from one account to another @param sessionContext @param txnAmount @param fromAccountNo @param toAccountNo @param legder @param txnRate @param messageType @return @throws FatalException	This operation provides the implementation of transferring funds in a particular  currency from one account to another account.   The funding account and the beneficiary account can be in different  currencies and can belong to different branches @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer from which the funds will be transferred.  @fcb.param MI,ISOBusinessContext,data, , BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.ATMBalanceResponse} containing the required information along with status. The transaction status will indicate the success of the transaction The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRE_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</li> <li>{@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIID}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOIID}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYNC_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FATAL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INSUFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_ACCOUNT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>NT_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	transferFunds	<p>This method is used for transferring funds from one account to another</p> <p>@param sessionContext @param txnAmount @param fromAccountNo @param toAccountNo @param legder @param txnRate @param messageType @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	transferFunds	This method is used for transferring funds from one account to another @param sessionContext @param txnAmount @param fromAccountNo @param toAccountNo @param legder @param txnRate @param messageType @return @throws FatalException	This method is used to transfer funds from the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for transfer funds @param txnAmount the transaction amount to be used for transfer funds @param currencyCode the currency code to be used @return TransactionStatus if transfer funds to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	transferFunds	This method is used for transferring funds from one account to another @param sessionContext @param txnAmount @param fromAccountNo @param toAccountNo @param legder @param txnRate @param messageType @return @throws FatalException	This method is used to transfer funds from the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param fromAccountNo the account id to be used for transfer funds @param toAccountNo the account id to be used for transfer funds @param txnAmount the transaction amount to be used for transfer funds @param serviceId the service id to be used @param txnRate the transaction rate to be used @param txnDate the transaction date @param messageType the messageType of the transaction @param eventId the event id to be used @param fromNarration the narration to be used for from account @param toNarration the narration to be used for to account @param cardNumber the cardNumber through which txn took place @return TransactionStatus if transfer funds to CASA failed or was success



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseDebitCash	This method is used for reversal of cash withdrawal for given account @param sessionContext @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt;</li></ul> <p>&lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;&lt;a href="#"&gt;@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseDebitCash	This method is used for reversal of cash withdrawal for given account @param sessionContext @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseDebitCash	This method is used for reversal of cash withdrawal for given account @param sessionContext @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICannelDem andDeposit Adapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \channel\ICannelDem andDepositAdapter	reverseDebi tCash	This method is used for reversal of cash withdrawal for given account @param sessionContext @return @throws FatalException	This method reverses Acquirer and Forwarder Fee charges to a CASA account for a transaction of Instrument.   After Debiting the Customer Account by Total Fee amount, Acquirer and Forwarder GL's are credited by Acquirer and Forwarder amount respectively. Acquirer and Forwarder Fee may or may not be present for the given transaction.  @fcb.param MI,SessionContext, sessionContext, , This parameter captures the basic information about the user of the system. @fcb.param MI,FinancialContext,txnValue, , Financial Context holds basic information about the transaction.  @fcb.param MI,String,accountId,1..16, The CASA account number, for which Cash deposit transaction has to be performed @fcb.param MI,double,transactionAmount, , The amount in terms of the transaction currency @fcb.param MI,int,mnemonic, , Mnemonic code of the transaction @fcb.param MI,int,processingCode, , The processing code under which this transaction will be   registered in the system @fcb.param MI,int,transactionType, , Can be one of the values from <ul> <li>{@link AbstractApplication#NORMAL_TRANSACTION}</li> <li>{@link AbstractApplication#ADVICE_TRANSACTION}</li> </ul> @fcb.param MI,TransactionStatus,status, , The status of the transaction which indicates success or failure @return BaseResponse @throws FatalException @throws OverridableException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseDebitCash	This method is used for reversal of cash withdrawal for given account @param sessionContext @return @throws FatalException	This method is used to reverse the debit of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service,it would contain the details of original transaction @return TransactionStatus will return status if reversal was successful
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseCreditCash	This method is used for reversal of cash deposit for given account @param sessionContext @return @throws FatalException	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRE_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li><a href="#">B_WDR_LIM</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#INV_ACCT_CODE</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#OLTP_INT_ERROR</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS</a></li> </ul> <p>@throws FatalException Please refer to the documentation above for <i>Transaction failure</i> section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseCreditCash	This method is used for reversal of cash deposit for given account @param sessionContext @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseCreditCash	This method is used for reversal of cash deposit for given account @param sessionContext @return @throws FatalException	This method is used to reverse the credit of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service,it would contain the details of original transaction @return TransactionStatus will return status if reversal was successful
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchCustomerRelationDetailForAccount	This method gets the customer/party details for given account number @param sessionContext @param accountId @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchCustomerRelationDetailForAccount	This method gets the customer/party details for given account number @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch CustomerRelationDetail of the customer account by calling the respective Party service @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for fetching the party details @return DDAPartyDetailsDTO will fetch the details of the party
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchCustomerRelationDetailForAccount	This method gets the customer/party details for given account number @param sessionContext @param accountId @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchPartyAddress	This method gets the customer/party address for given party id @param sessionContext @param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchPartyAddress	This method gets the customer/party address for given party id @param sessionContext @param partyId @return @throws FatalException	This method is used to fetching address details of the customer account by calling the respective Party service @param sessionContext the session context to be used in interacting with the service @param partyId the party id to be used for fetching address @return PartyAddressResponse will return the address of the party
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	adHocStatementRequest	This method requests for a adhoc statement for particular account @param sessionContext @param accountId @param period @param deliveryMode @return @throws FatalException	This operation gives the status of adhoc statement request It takes the period,mode of delivery as input    These details are returned as a data transfer object of type DDABalancedTO  @fcb.param MI,ATMChannelContext,channelContext,ATMChannel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate. etc  @return ATMBalanceResponse A response object which holds the details of the  The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>     <b>Transaction failure</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIID}</li> </ul>   Error Reason may be one of the following <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}</li> <li>{@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	adhocStatementRequest	This method requests for a adhoc statement for particular account @param sessionContext @param accountId @param period @param deliveryMode @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	adhocStatementRequest	This method requests for a adhoc statement for particular account @param sessionContext @param accountId @param period @param deliveryMode @return @throws FatalException	This method is used to issue adhoc statement of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for adhoc request @return TransactionStatus if adhoc request logging to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchDefaultAddressType	This method fetches the default address of account @param sessionContext @param accountId @param accountType @throws FatalException	This operation allows for cheque book request for number of cheque leaves to be issued,no of cheque book to be issued     These details are returned as a data transfer object of type ATMBalanceResponse  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_N T_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OU TSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B AL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchDefaultAddressType	This method fetches the default address of account @param sessionContext @param accountId @param accountType @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchDefaultAddressType	This method fetches the default address of account @param sessionContext @param accountId @param accountType @throws FatalException	This method is used to issue adhoc statement of the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for adhoc request @return TransactionStatus if adhoc request logging to CASA failed or was success
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchDefaultAddressType	This method fetches the default address of account @param sessionContext @param accountId @param accountType @throws FatalException	This method fetches the default address of account @param sessionContext the session context to be used in interacting with the service @param accountId the account id to fetch default address @param accountType the account type of the request @throws FatalException
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseTransferFundsToBeneficiaries	This method is used for reversal of fund transfer for given account @param sessionContext @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	reverseTransferFundsToBeneficiaries	This method is used for reversal of fund transfer for given account @param sessionContext @return @throws FatalException	This method is used for reversal of fund transfer for given account @param sessionContext the session context to be used in interacting with the service @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchAccountStatus	This method checks if the account status is valid CASA account @param sessionContext @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchAccountStatus	This method checks if the account status is valid CASA account @param sessionContext @return @throws FatalException	This method checks if the account status is valid CASA account @param sessionContext the session context to be used in interacting with the service @param accountId the account number to check for restriction services @param serviceId the service id of the request @param eventId the event id of the service @return @throws FatalException
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	checkKYCforParty	This method checks if party is KYC unconfirmed or blacklisted @param sessionContext @param partyId @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	checkKYCforParty	This method checks if party is KYC unconfirmed or blacklisted @param sessionContext @param partyId @return @throws FatalException	This method checks if party is KYC unconfirmed or blacklisted @param sessionContext the session context to be used in interacting with the service @param partyId the party id against which KYC needs to be checked @return @throws FatalException
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchAccountDetails	This method is used to get the customer account details by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for inquire @param txnAmount the transaction amount to be used for inquire @param currencyCode the currency code to be used @return	This operation should be use to inquire upon the statement of  accounts for a CASA account. The operation fetches all the transactions   done on a particular account in the specified period. This includes both  customer-initiated transactions and system initiated transactions like  interest and SC capitalization.   The statement of accounts for the current period fetches the details of all  transactions conducted during the current statement period. Similarly the  previous statement period fetches all the transactions performed during  the



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			ddaBalanceDTO will fetch the details after inquire of account	<pre> previous statement period.&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill payment transaction.&lt;br&gt; @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.&lt;br&gt; @return ATMStatementResponse The response contains details of the all transactions as well as&lt;br&gt; the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; follwing will be in the response&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS} &lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_I NVALID_USERID}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_BANK}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_C ORRDFN_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_DUM MY_ERROR_CODE}&lt;/li&gt; &lt;/ul&gt; @throws FatalException See the return paragraph above </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchAccountDetails	This method is used to get the customer account details by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for inquire @param txnAmount the transaction amount to be used for inquire @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after inquire of account	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchAccountDetails	This method is used to get the customer account details by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for inquire @param txnAmount the transaction amount to be used for inquire @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after inquire of account	This method is used to get the customer account details by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for inquire @param txnAmount the transaction amount to be used for inquire @param currencyCode the currency code to be used @return ddaBalanceDTO will fetch the details after inquire of account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	transferFundsToBeneficiary	This method is used to transfer funds from the customer account by calling the respective CASA service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for transfer funds @param txnAmount the transaction amount to be used for transfer funds @param currencyCode the currency code to be used @return TransactionStatus if transfer funds to CASA failed or was success	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchVeryBasicPartyDetails	This method fetches the very basic details about the party like name,accounttype @param sessionContext @param accountId @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchVeryBasicPartyDetails	This method fetches the very basic details about the party like name,accounttype @param sessionContext @param accountId @return @throws FatalException	This method fetches the very basic details about the party like name,accounttype @param sessionContext @param accountId @return @throws FatalException
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchVeryBasicPartyDetails	This method fetches the very basic details about the party like name,accounttype @param sessionContext @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation gives the details of the CASA account balances such&lt;br&gt; as the available balance, the uncleared balance and the earmarked balance &lt;br&gt; using Balance Enquiry&lt;br&gt; These details are returned as a data transfer object of type DDABalanceDTO&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate. etc&lt;br&gt; @return ATMBalanceResponse A response object which holds the details of the&lt;br&gt; balances for the given account.The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIDD}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt;  </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> i&gt; &lt;li&gt; {@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt; {@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation allows for cheque book request for number of cheque leaves to be issued, no of cheque book to be issued &lt;br&gt; &lt;br&gt; These details are returned as a data transfer object of type ATMBalanceResponse&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATMChannel Context which must contain the ATM Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOID}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation allows for cheque book request for number of cheque leaves to be issued,no of cheque book to be issued &lt;br&gt; &lt;br&gt; These details are returned as a data transfer object of type ATMBalanceResponse&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_N T_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OU TSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B AL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/l i&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;</li> </ul> <p>&lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation should be use to inquire upon the statement of&lt;br&gt; accounts for a CASA account. The operation fetches all the transactions &lt;br&gt; done on a particular account in the specified period. This includes both&lt;br&gt; customer-initiated transactions and system initiated transactions like&lt;br&gt; interest and SC capitalization. &lt;br&gt; The statement of accounts for the current period fetches the details of all&lt;br&gt; transactions conducted during the current statement period. Similarly the&lt;br&gt; previous statement period fetches all the transactions performed during&lt;br&gt; the previous statement period.&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>payment transaction.&lt;br&gt; @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.&lt;br&gt; @return ATMStatementResponse The response contains details of the all transactions as well as&lt;br&gt; the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; following will be in the response&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_I NVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_BANK}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_C ORRDFN_NOT_FOUND}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_DUM MY_ERROR_CODE}&lt;/li&gt; &lt;/ul&gt; @throws FatalException See the return paragraph above</p>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.&lt;br&gt; &lt;ol&gt; It validates following channels against banking facilities.&lt;br&gt; &lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt; &lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt; &lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt; &lt;li&gt;POS_CHANNEL = "POS" -</p>	<p>This operation gives the status of adhoc statement request It takes the period,mode of delivery as input&lt;br&gt; &lt;br&gt; These details are returned as a data transfer object of type DDABalancedDTO&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext,ATMChannel Context which must contain the ATM Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>hasPOS&lt;/li&gt; &lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt; &lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</p>	<p>rate. etc&lt;br&gt; @return ATMBalanceResponse A response object which holds the details of the&lt;br&gt;The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_N T_FLG_CLS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OU TSTANDING_PO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B AL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark force purchase against a ATM Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	<p>This method validates if account has the facility on the channel available.</p> <p>&lt;ol&gt; It validates following channels against banking facilities.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking&lt;/li&gt;</li> <li>&lt;li&gt;TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;MOBILE_BANKING = "MB" - hasMobileBanking&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "POS" - hasPOS&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "ATM" - hasATM&lt;/li&gt;</li> <li>&lt;li&gt;POS_CHANNEL = "BRN"&lt;/li&gt;</li> </ul>	<p>This operation enables you to perform a earmark force purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	This method validates if account has the facility on the channel available. <ul style="list-style-type: none"> <li>It validates following channels against banking facilities. <ul style="list-style-type: none"> <li>INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking</li> <li>TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS</li> <li>MOBILE_BANKING = "MB" - hasMobileBanking</li> <li>POS_CHANNEL = "POS" - hasPOS</li> <li>POS_CHANNEL = "ATM" - hasATM</li> <li>POS_CHANNEL = "BRN"</li> </ul> </li> </ul>	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateChannelFacility	This method validates if account has the facility on the channel available. <ul style="list-style-type: none"> <li>It validates following channels against banking facilities. <ul style="list-style-type: none"> <li>INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking</li> <li>TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS</li> <li>MOBILE_BANKING = "MB" - hasMobileBanking</li> <li>POS_CHANNEL = "POS" - hasPOS</li> <li>POS_CHANNEL = "ATM" - hasATM</li> <li>POS_CHANNEL = "BRN"</li> </ul> </li> </ul>	This method validates if account has the facility on the channel available. <ul style="list-style-type: none"> <li>It validates following channels against banking facilities. <ul style="list-style-type: none"> <li>INTERNATE_BANKING_CHANNEL = "IB" - hasTeleBanking</li> <li>TELEBANKING_IVR_CHANNEL = "IVR" - hasPOS</li> <li>MOBILE_BANKING = "MB" - hasMobileBanking</li> <li>POS_CHANNEL = "POS" - hasPOS</li> <li>POS_CHANNEL = "ATM" - hasATM</li> <li>POS_CHANNEL = "BRN"</li> </ul> </li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS} &lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark force purchase against a ATM Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	<p>This operation enables you to perform a earmark force purchase against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EX CEP_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold defination @param holdDefintionId @return	
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold defination @param holdDefintionId @return	This method fetches the description for a particular hold defination @param holdDefintionId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchEarmarkDescription	This method fetches the description for a particular hold definition @param holdDefintionId @return	This method fetches the description for a particular hold definition @param sessionContext the session context to be used in fetching @param earmarkParametersId the earmark paramater id to be used for fetching the description @return
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchFundsManagementInquiryResponse	This method fetches the funds management inquiry response @param sessionContext @param instructionSearchDTO @return @throws FatalException	This method fetches the Information log of completed transactions and standing instructions on the basis of basic search criteria i.e Party Id(Mandatory), FromDate(Mandatory), ToDate(Mandatory), Account Id(Optional), PaymentType(Optional) @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI, com.ofss.fc.app.channel.dto.log.PaymentSearchDTO @return {@link com.ofss.fc.app.channel.dto.log.InformationLogInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	fetchFundsManagementInquiryResponse	This method fetches the funds management inquiry response @param sessionContext @param instructionSearchDTO @return @throws FatalException	
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateAccountStatus	This method checks if the account status is not blocked,closed,dormant @param sessionContext @param accountId @param serviceId @param eventId @throws FatalException	This operation allows for cheque book request for number of cheque leaves to be issued,no of cheque book to be issued     These details are returned as a data transfer object of type ATMBalanceResponse  @fcb.param MI,ATMChannelContext,channelContext, , ATM

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return ATMBalanceResponse A response object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li><b>Success</b></li> <li><a href="#">com.ofss.fc.enumeration.ResponseCodeType#SUCCESS</a></li> <li><b>Transaction failure</b></li> <li><a href="#">com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D</a></li> </ul> <p>Error Reason may be one of the following</p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_N T_FLG_CLS</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#NO_OU TSTANDING_PO</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY</a></li> <li><a href="#">com.ofss.fc.enumeration.ReplyMessageType#ACCT_B AL_NT_ZERO</a></li> <li><a href="#">ErrorConstants#MID_ECH_INVALID_ACCT_NO</a></li> <li><a href="#">ErrorConstants#MID_EUT_INVALID_USERID</a></li> <li><a href="#">ErrorConstants#MID_EST_INV_BANK</a></li> </ul> <p>@throws FatalException Please refer to the documentation above for Transaction failure section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateAccountStatus	This method checks if the account status is not blocked,closed,dormant @param sessionContext @param accountId @param serviceId @param eventId @throws FatalException	This operation should be use to inquire upon the statement of  accounts for a CASA account. The operation fetches all the transactions   done on a particular account in the specified period. This includes both  customer-initiated transactions and system initiated transactions like  interest and SC capitalization.   The statement of accounts for the current period fetches the details of all  transactions conducted during the current statement period. Similarly the  previous statement period fetches all the transactions performed during  the previous statement period.  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill payment transaction.  @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.  @return ATMStatementResponse The response contains details of the all transactions as well as  the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure  <b>Transaction Success</b> follwing will be in the response  <ul> <li>{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS}</li> </ul> <b>Transaction Failure</b> Error Reason may be one of the following <ul> <li>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_INVALID_USERID}</li> <li>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_INV_BANK}</li> <li>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_CORRDFN_NOT_FOUND}</li> <li>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				NV_ACCT_NO}</li> <li>{@link com.ofss.fc.common.FAPIErrorConstants#MID_DUMMY_ERROR_CODE}</li> </ul> @throws FatalException See the return paragraph above
IChannelDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateAccountStatus	This method checks if the account status is not blocked,closed,dormant @param sessionContext @param accountId @param serviceId @param eventId @throws FatalException	This operation gives the status of adhoc statement request It takes the period,mode of delivery as input    These details are returned as a data transfer object of type DDABalanceDTO  @fcb.param MI,ATMChannelContext,channelContext,ATMChannel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate. etc  @return ATMBalanceResponse A response object which holds the details of the  The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>     <b>Transaction failure</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOID}</li> </ul> Error Reason may be one of the following <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_NT_FLG_CLS}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OUTSTANDING_PO}</li> <li>{@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BAL_NT_ZERO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK}&lt;/li&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateAccountStatus	This method checks if the account status is not blocked,closed,dormant @param sessionContext @param accountId @param serviceId @param eventId @throws FatalException	
IChannelDemandDeposit Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelDemandDepositAdapter	validateAccountStatus	This method checks if the account status is not blocked,closed,dormant @param sessionContext @param accountId @param serviceId @param eventId @throws FatalException	This method checks if the account status is not blocked,closed,dormant @param sessionContext the session context to be used in processing @param accountId the account id against which restriction needs to be checked @param serviceId the service code to be used @param eventId the event code to be used @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceLoanAdapter	checkIfAccountsAreClosed		This is the helper method to check whether a particular BranchParameters record can be closed or not.  If the BranchParameters can be closed, it returns true, else it returns false. This method checks whether any account(s) is (are) opened under any domain category TD, CASA, Loan,  If no account is opened , the branch can be closed, otherwise, we should not allow to close the branch. @param sessionContext : {@link SessionContext} @param branchCode : String @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceLoanAdapter	checkIfAccountsAreClosed		This method returns true, if all Loan accounts are closed under the given branch else returns false.
IWorkflowTrackerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\workflow\IWorkflowTrackerAdapter	collectData		This method will execute the workflow tracker search for the common attributes. @param searchText @param compositeName @param pageInformationDTO @return @throws FatalException
IWorkflowTrackerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\workflow\IWorkflowTrackerAdapter	collectData		This method will execute the workflow tracker search for custom attributes. It will call the workflow tracker search handler specified in the {@link WorkflowTrackerSearchConfiguration} to execute the search. @param configuration @param searchText @param pageInformationDTO @return
IWorkflowTrackerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\workflow\IWorkflowTrackerAdapter	collectData		This method will call the search handler specified in the input to execute the search.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	updateDdAccountPreferences	<p>This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,a ccountConfigurationDTO, , This class represents the DTO class for demand deposit account &lt;br&gt; configuration.This class contains information about the preferences given for the demand deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	<p>@param sessionContext @param inputData @param oDAccountConfigurationDTO @param offerReferenceArray @param accountId @param ddAdapter @param i @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	updateDdAccountPreferences	This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This method is used to update the Account Configurations for a Demand Deposit Account that has already been opened previously. Account level preferences such as statement preferences, minimum balance, debit card preferences, account facilities such as sweep in sweep put instructions, etc. are updated. Additionally, operating instructions both financial and non financial for the account, cheque book requests and default communication address for the account are also updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	validateAccount	This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This method is used to validate account opening or first disbursement date at the account / product group level @param sessionContext @param prodGrpLinkage @param creditFacility @param accountOpeningOrSettlementPayoutDate @param applicationBranch @return validationErrors @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	validateAccount	This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,a ccountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This method is used to validate account opening or first disbursement date at the application level @param sessionContext @param productGroupLinkages @param creditFacility @param accountOpeningOrSettlementPayoutDate @param applicationBranch @return validationErrors @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	validateAccount	<p>This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account &lt;br&gt; configuration.This class contains information about the preferences given for the demand deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	<p>This method is used to validate the account opening or first disbursement date against the data basis type date (structure solution / offer acceptance date) @param sessionContext @param prodGrpLinkage @param creditFacility @param accountOpeningOrSettlementPayoutDate @return validationErrors @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	validateAccount	This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This method is used to validate the Account before assigning an Operating Instruction to it. @param sessionContext @param accountId @param serviceId @return boolean isServiceRestricted @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	validateAccount	This method is used to call the DDA module to update the Account Preferences for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information	This method is used to open a Lending Account. The following validations are performed:  1. There should not be any warning indicators for the parties applying for the PG  2. The KYC check should be completed for all the parties  3. All the preliminary conditions should have been complied with   4. Productgroup level Origination fees should have been collected   5. Application

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,a ccountConfigurationDTO, , This class represents the DTO class for demand deposit account &lt;br&gt; configuration.This class contains information about the preferences given for the demand deposit account&lt;br&gt; as well as preferences captured for each account holder.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	<p>should be verified successfully.&lt;br&gt; Once all the validations are successful, the facility is confirmed in LCM and the account is created &lt;br&gt; The account id is then updated in Origination&lt;br&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @param MI,FacilityApprovalDataDTO facilityApprovalDataDTO - the facility level approval data @return LendingAccountCreationResponse, this class returns the LendingAccountCreationResponse object, @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This check is part of Pre-create account validation. &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties &lt;br&gt; applying to the product group for which account is being opened. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>account opening are present for applicants applying to the products.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#CONFIGURATIONS_DO_NOT_EXIST} : FC_OR_AC_0001 &lt;br&gt; This error occurs when account configuration for an offer is not found &lt;br&gt; {@link OriginationErrorConstants#OR_PRE_ACCOUNT_CREATION_VALIDATIONS_FAILED} : FC_OR_AC_0052 &lt;br&gt; This error occurs as a part of pre account creation checks. &lt;br&gt; 1. {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; 2. {@link OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This exception is thrown if KYC has not been confirmed for all parties applying to the product group for which account is being opened. &lt;br&gt; 3. {@link OriginationErrorConstants#PRELIMINARY_CONDITIONS_NON_COMPLIANCE} : FC_OR_OF_0057 &lt;br&gt; Preliminary conditions for the borrowing entity have not been complied .&lt;br&gt; 4. {@link OriginationErrorConstants#OR_UNCOLLECTED_ORIGINATION_FEES} : FC_OR_OF_0051 &lt;br&gt; The product group level pricing fees have not been collected .&lt;br&gt; 5. {@link OriginationErrorConstants#OR_OFFER_APP_NOT_VERIFIED} : FC_OR_OF_0047 &lt;br&gt; Application has not been verified yet &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_LENDING_APPLICATION_INVALID} : FC_OR_ST_0001 &lt;br&gt; This error occurs as a part of state transition checks when the application is not a lending application&lt;br&gt; &lt;br&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchAccountDetails	This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This method merges existing account preferences with the preferences captured during account configuration @param newPreferences @param existingPreferences @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchAccountDetails	This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This service is used to update the Account Holder preferences for an already existing demand deposit account.  In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,DDAccountHolderPreferenceDTO,preferenceDTO, , This class contains information about preferences captured for the applicant of the demand deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, demand deposit statement delivery preferences and debit card preferences such as color , type, etc. @fcb.param MI, Boolean, isUpdateAccountRelationShipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchAccountDetails	<p>This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.&lt;br&gt; @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.&lt;br&gt; @throws FatalException</p>	<p>This method is used to log event related to Account creation of demand deposit. @param sessionContext @param submissionId @param applicationId @param response @param prodGrpLinkage</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAadapter	fetchAccountDetails	This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This method is used to call the DDA module to validate the Account Id for a Demand Deposit Account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException
IOriginationDDAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAadapter	fetchAccountDetails	This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This method is used to call the DDA module to get the details for an account that has already been opened previously. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,Account Id : This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchDDPreferences	This service is used for inquiring the latest casa preferences for the account provided as an input from CASA module @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,String, accountId , , This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This service will fetch account configurations done for a CASA account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g: Internet banking, Debit Card, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the account configuration details for demand deposit products. @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchChequeBookRequest	This service will inquire cheque book requests made for the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,String, accountId , , This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This service will inquire cheque book requests made for the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,String, accountId , , This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchChequeBookRequest	This service will inquire cheque book requests made for the account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,String, accountId , , This represents the Account Id of the Demand Deposit Account already opened.  @throws FatalException	This service will fetch account configurations done for a CASA account opened. In case of joint applications where joint applicant is not present, the preferences that are selected by this joint applicant is based on what preferences are enabled for the account. For e.g: Internet banking, Debit Card, etc. if enabled at account level can be opted by the joint applicant. Preferences can be updated even after an account is opened; hence latest preferences are fetched always. In case of any exception, ignore and move ahead. The details will be fetched from the draft version of account in case of such exceptions. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the account configuration details for demand deposit products. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAAdapter	constructDDAccountCreationDTO	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	constructODAccountCreationDTO	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO,, This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	@param sessionContext @param inputData @param methodName @param oDAccountConfigurationDTO @param approvedTenor @param applicationId @param campaignCode @param offerReferenceArray @param simulationId @param facilityId @param ddAdapter @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	constructODAccountCreationDTO	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method will construct the Over Draft account creation DTO from the input Overdraft account configuration DTO @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ODAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for overdraft account configuration..  @return OverdraftAccountCreationDTOResponse, this class returns the OverdraftAccountCreationDTOResponse object, containing the OverdraftAccountCreationDTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	constructODAccountCreationDTO	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method is used to call the DDA module to set the account configuration DTO for OverDraft account @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	openODAccount		@param sessionContext @param inputData @param methodName @param oDAccountConfigurationDTO @param approvedTenor @param applicationId @param campaignCode @param offerReferenceArray @param simulationId @param facilityId @param ddAdapter @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	addAccountFacilityLinkage	This method is used to add account facility linkage for an existing CASA account that has overdraft facility @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,a ccountConfigurationDTO,, This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	@param sessionContext @param inputData @param oDAccountConfigurationDTO @param approvedTenor @param facilityId @param ddAdapter @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	addAccountFacilityLinkage	This method is used to add account facility linkage for an existing CASA account that has overdraft facility @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method is used to add account facility linkage for an existing CASA account that has overdraft facility @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAAdapter	openDDAccount	This method is used to call the DDA module to open a Demand Deposit Account . @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method is used to call the DDA module to open a Demand Deposit Account . @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @fcb.param MI, String, applicatioId - the deposit application ID @fcb.param MI, String, borrowingEntityId This field denotes the borrowing entity applying for product groups within this lending application.  For all products groups applied by same borrowing entity, they are grouped under same lending application.  Borrowing entities are created and stored in LCM system. @fcb.param MI, String, campaignCode This attribute indicates the campaign code used in the application @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException
IOriginationDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAAdapter	openDDAccount	This method is used to call the DDA module to open a Demand Deposit Account . @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,a	This service will create an demand deposit account for given account configuration. It is wrapper service called from process. It does the following in sequence 1.Calls the DDAOpeningApplicationService to - populate the DDA account opening DTO - open a DD account The account id is set in the response. 2. Adds the account reference to the corresponding offer in the application in origination. 3. Performs state transition for the product group and the application , if

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>ccountConfigurationDTO, , This class represents the DTO class for demand deposit account configuration. This class contains information about the preferences given for the demand deposit account as well as preferences captured for each account holder. @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException</p>	<p>necessary @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, submissionId, , Indicates the identifier for the submission @fcb.param MI,String, applicationId, , Indicates the identifier for the application @fcb.param MI, int, productGroupSerialNumber, , Indicates the serial number position at which the product group is available in the application. @fcb.param MI, int, offerSerialNumber, , Indicates the offer serial number @fcb.param MI,DDAccountConfigurationDTO, ddaAccountConfigurationDTO, , This class holds account configuration details for the demand deposit offer. It contains information about the accounts being configured within the simulation such as statement preferences, card preferences, etc. @return {@link DDAccountOpeningResponse} This class contains the account opening response i.e. account id @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_DEPOSIT_APPLICATION_INVALID} :FC_OR_ST_0002 &lt;br&gt; This exception occurs during invalid state transition where the application is not a Deposit application&lt;br&gt; &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	openDDAccount	This method is used to call the DDA module to open a Demand Deposit Account . @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException	This method is used to call the DDA module to open a Demand Deposit Account . @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @fcb.param MI, String, applicatiId - the deposit application ID @fcb.param MI, String, borrowingEntityId This field denotes the borrowing entity applying for product groups within this lending application.  For all products groups applied by same borrowing entity, they are grouped under same lending application.  Borrowing entities are created and stored in LCM system. @fcb.param MI, String, campaignCode This attribute indicates the campaign code used in the application @return DdCreationResponseDTO, this class returns the DDAccountOpeningResponse object, along with the TransactionStatus for the account preferences update. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchExistingAccountApplicableForOD	This method would call the DDA module and fetch the CASA accounts with the mentioned party-ids and offer-ids and having Account-status as Regular and not linked to any OD facility @fcb.param MI , List<String> partyIds, List of party-ids for which existing CASA account is to be fetched @fcb.param MI , List<String> offerIds, List of offer-ids for which existing CASA account is to be fetched @return ExistingCASAInquiryDataDTO[], Array of existing CASA accounts for which OD facility is not present @throws FatalException	This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility. An existing demand deposit account can be used only if the following conditions are fulfilled:    Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners.    <b>For e.g.</b> : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list.     Also, existing CASA selected, must not be linked to any other Overdraft facility.    Refactored to this module from {@link LendingApplicationService} @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @fcb.param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception   {@link DemandDepositAccountErrorConstant#DDA_OFFER_INVALID} : FC_OR_DE_0081   This error occurs when:  The offer details are being fetched for incorrect offer identifier or offer has expired.   

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchExistingAccountsApplicableForOD	This method would call the DDA module and fetch the CASA accounts with the mentioned party-ids and offer-ids and having Account-status as Regular and not linked to any OD facility @fcb.param MI, List<String> partyIds, List of party-ids for which existing CASA account is to be fetched @fcb.param MI, List<String> offerIds, List of offer-ids for which existing CASA account is to be fetched @return ExistingCASAInquiryDataDTO[], Array of existing CASA accounts for which OD facility is not present @throws FatalException	This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility.  An existing demand deposit account can be used only if the following conditions are fulfilled:     Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners.     <b>For e.g.</b> : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list.     Also, existing CASA selected, must not be linked to any other Overdraft facility.     @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @fcb.param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception   {@link OriginationErrorConstants#OR_OFFER_INVALID} : FC_OR_DE_0081   This error occurs when:  The offer details are being fetched for incorrect offer identifier or offer has expired.   



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	fetchExistingAccountApplicableForOD	This method would call the DDA module and fetch the CASA accounts with the mentioned party-ids and offer-ids and having Account-status as Regular and not linked to any OD facility @fcb.param MI , List<String> partyIds, List of party-ids for which existing CASA account is to be fetched @fcb.param MI , List<String> offerIds, List of offer-ids for which existing CASA account is to be fetched @return ExistingCASAInquiryDataDTO[], Array of existing CASA accounts for which OD facility is not present @throws FatalException	This method would call the DDA module and fetch the CASA accounts with the mentioned party-ids and offer-ids, having Account-status as Regular and not linked to any OD facility @fcb.param MI , List<String> partyIds, List of party-ids for which existing CASA account is to be fetched @fcb.param MI , List<String> offerIds, List of offer-ids for which existing CASA account is to be fetched @return ExistingCASAInquiryDataDTO[], Array of existing CASA accounts for which OD facility is not present @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	checkIfODAccountIsRegulated	This method will call DDAPreferencesApplicationService with given facts to check if the account configuration is a regulated account configuration @param factInformationDTOs @return boolean	
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	checkIfODAccountIsRegulated	This method will call DDAPreferencesApplicationService with given facts to check if the account configuration is a regulated account configuration @param factInformationDTOs @return boolean	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	checkIfODAccountIsRegulated	This method will call DDAPreferencesApplicationService with given facts to check if the account configuration is a regulated account configuration @param factInformationDTOs @return boolean	This method will call DDAPreferencesApplicationService with given facts to check if the account configuration is a regulated account configuration @param factInformationDTOs @return boolean
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	levyFeesOnAccountConfiguration	On OD account opening, If settlement instruction is not present , Origination workflow callThis service to <ol> <li>Deduct LMI and Legal fees.:These fees are not maintained in pricing.They are miscellaneous debits for DDA module. <li>Levy deffered fees which are marked in account opening. - {@link #levyDeferredFeesOnAccountOpening(String)}. </ol>   If OD settlement is present then all the fees will be deducted in settlement execution.	On OD account opening, If settlement instruction is not present , Origination workflow callThis service to <ol> <li>Deduct LMI and Legal fees.:These fees are not maintained in pricing.They are miscellaneous debits for DDA module. <li>Levy deffered fees which are marked in account opening. - {@link #levyDeferredFeesOnAccountOpening(String)}. </ol>   If OD settlement is present then all the fees will be deducted in settlement execution.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	levyFeesOnAccountConfiguration	On OD account opening, If settlement instruction is not present , Origination workflow callThis service to <ol> <li>Deduct LMI and Legal fees.:These fees are not maintained in pricing.They are miscellaneous debits for DDA module. <li>Levy defferred fees which are marked in account opening. - {@link #levyDeferredFeesOnAccountOpening(String)}. </ol>   If OD settlement is present then all the fees will be deducted in settlement execution.	@param sessionContext @param accountId @param LMI @param legal @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	levyFeesOnAccountConfiguration	On OD account opening, If settlement instruction is not present , Origination workflow callThis service to <ol> <li>Deduct LMI and Legal fees.:These fees are not maintained in pricing.They are miscellaneous debits for DDA module. <li>Levy defferred fees which are marked in account opening. - {@link #levyDeferredFeesOnAccountOpening(String)}. </ol>   If OD settlement is present then all the fees will be deducted in settlement execution.	On OD account opening, If settlement instruction is not present , Origination workflow call this service to <ol> <li>Deduct LMI and Legal fees.:These fees are not maintained in pricing.They are miscellaneous debits for DDA module. <li>Levy defferred fees which are marked in account opening. - {@link #levyDeferredFeesOnAccountOpening(String)}. </ol>   If OD settlement is present then all the fees will be deducted in settlement execution.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	amendODAccount		This class is used to create over draft account @param sessionContext @param inputData @return AccountOpeningDetailsDTO @throws FatalException
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	amendODAccount		
IOriginationDDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDDAdapter	updateDDAccountHolderPreferences		This service is used to update the Account Holder preferences for an already existing demand deposit account.  In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>preferences are being updated. @fcb.param MI,DDAccountHolderPreferenceDTO,preferenceDTO, , This class contains information about preferences captured for the applicant of the demand deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, demand deposit statement delivery preferences and debit card preferences such as color , type, etc. @fcb.param MI, Boolean, isUpdateAccountRelationShipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>
IHardshipProcessForApprovalCallingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IHardshipProcessForApprovalCallingAdapter	callApproval	@param HardshipProcessForApprovalRequestDTO @return TransactionStatus @throws FatalException	@param HardshipProcessForApprovalRequestDTO @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingPartyAdapter	fetchAllAccountsOfParty	This method will fetch array of {@link PartyAccountInformationDTO} which contains account related information about given Party @param sessionContext This parameter captures the basic information about the user of the system. @param partyId Identifier of Party for which account information needs to be found out. @return Array of {@link PartyAccountInformationDTO} which contains account related information about given Party @throws com.ofss.fc.infra.exception.FatalException	This method processes particular addition done in Price Policy Chart @param sessionContext the session context @param pricePolicyChartAddition {@link PricePolicyChartAddition} to be processed @throws FatalException
IPricingParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingPartyAdapter	fetchAllAccountsOfParty	This method will fetch array of {@link PartyAccountInformationDTO} which contains account related information about given Party @param sessionContext This parameter captures the basic information about the user of the system. @param partyId Identifier of Party for which account information needs to be found out. @return Array of {@link PartyAccountInformationDTO} which contains account related information about given Party @throws com.ofss.fc.infra.exception.FatalException	This method will fetch array of {@link PartyAccountInformationDTO} which contains account related information about given Party @param sessionContext This parameter captures the basic information about the user of the system. @param partyId Identifier of Party for which account information needs to be found out. @return Array of {@link PartyAccountInformationDTO} which contains account related information about given Party @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.pricing.IPricingPartyAdapter#fetchAllAccountsOfParty(com.ofss.fc.app.context.SessionContext, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			ption	
IPricingParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingPartyAdapter	fetchAllAccountsOfParty	This method will fetch array of {@link PartyAccountInformationDTO} which contains account related information about given Party @param sessionContext This parameter captures the basic information about the user of the system. @param partyId Identifier of Party for which account information needs to be found out. @return Array of {@link PartyAccountInformationDTO} which contains account related information about given Party @throws com.ofss.fc.infra.exception.FatalException	@param context @param partyId @param modifiedFactMap @return @throws FatalException
IPricingParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingPartyAdapter	getPrimaryPartyForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns the details of primary party for the account @param accountId identifier of an account. @return String account ID.	@see IPricingAccountAdapter#getBasicDetailsForAccount(String, String)
IPricingParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingPartyAdapter	getPrimaryPartyForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns the details of primary party for the account @param accountId identifier of an account. @return String account ID.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyExpiryDateChange	This method is to notify CASA on change of expiry Date @param accountIdList @param facilityId @param expiryDate @param facilityType @throws FatalException	This method updates the FacilityExpiryEvent details in EOD/BOD logger tables and also notifies the CASA regarding the expiry date related changes. @param newFacility @param prevFacilityRecord @param entityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityCreation	This method is to notify CASA when a uncleared facility or Temp excess Facility is created @param facility	This method is used to notify the CASA when the creation of the facility happens @param facility @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnLinkage	This method is to notify CASA when a uncleared facility or Temp Excess Facility is linked during the batch @param facility	This method is used to notify CASA on the linkage of the new facility. Since the linkage is initiated from LCM the notification is being sent to CASA. This is not applicable for Credit Facility as it is triggered through the module. @param facilityDtls @param facilityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnLinkage	This method is to notify CASA when a uncleared facility or Temp Excess Facility is linked during the batch @param facility	This method is used to notify CASA on the linkage of the new facility. Since the linkage is initiated from LCM the notification is being sent to CASA. This is not applicable for Credit Facility as it is triggered through the module. @param facilityDtls @param facilityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnBorrowingEntityMergerAction	This method is to notify other modules when ever t BorrowingEntityMerger happens @param oldBorrowingEntity @param newBorrowingEntity	This method is to update utilization amount in corresponding borrowing entities on change of borrowing entity @param srcBorrowingEntity @param destBorrowingEntity @param facilityId @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnBorrowingEntityMergerAction	This method is to notify other modules when ever t BorrowingEntityMerger happens @param oldBorrowingEntity @param newBorrowingEntity	This method is used to notify the account modules for the borrowing entity change. @param sourceBorrowingEntity @param destBorrowingEntity



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityExpiry	This method is to notify DDA when a uncleared facility or Temp Excess Facility is expired @param accountIdList @param facilityType @throws FatalException	This method is used for expiry processing. @param entityId @param fromString @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityClosure	This method is used to notify the CASA on the closure of the facility. @param accountIdList @param facilityId @param facilityType @param expiryDate @throws FatalException	This method is used to notify the CASA when the creation of the facility happens @param facility @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityLimitChange	This method is used to notify the CASA on the limit change. @param facilityNotificationDTO @throws FatalException	This method is used to check whether notification is required for the CASA due to the facility update. The fields that needs to be checked are the limit amount and the facility expiry date. The rest of the fields are non modifiable and sanctioned amount is not required by CASA. If there are changes in the values , the notification method will be invoked. Since facility expiry date needs to call a separate method the check for this will not be done in this method. @param newObject @param facilityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityLimitChange	This method is used to notify the CASA on the limit change. @param facilityNotificationDTO @throws FatalException	This method executes the FacilityAmountSchedule Processing taking arguments as appContext and batchData @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityLimitChange	This method is used to notify the CASA on the limit change. @param facilityNotificationDTO @throws FatalException	This method executes the FacilityAvailmentExpiry Processing service taking arguments as appContext and batchData

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityLimitChange	This method is used to notify the CASA on the limit change. @param facilityNotificationDTO @throws FatalException	This method executes the FacilityAmountScheduleExpiry Processing taking arguments as appContext and batchData
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnFacilityLimitChange	This method is used to notify the CASA on the limit change. @param facilityNotificationDTO @throws FatalException	This method is used to check whether notification is required for the CASA due to the facility update. The fields that needs to be checked are the limit amount and the facility expiry date. The rest of the fields are non modifiable and sanctioned amount is not required by CASA. If there are changes in the values , the notification method will be invoked. Since facility expiry date needs to call a separate method the check for this will not be done in this method. @param oldObject @param newObject @param facilityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnLimitAmountDueToUnAvailability	This method is used to notify the CASA on the limit change. This is called during the change in the effective limit amount due to a Credit transaction when the facility is unavailable. @param facilityNotificationDTO @throws FatalException	This method is used to notify CASA on the limit change. This is applicable only in case of unavailable facility getting a credit. This is because in LCM the effective limit amount of the unavailable facility will be reduced to the extent of its utilization. Hence there is a limit change on a Credit and CASA needs to be notified. @param facilityId @param effectiveValueDate @param facilityType @throws FatalException
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnScheduleChange	This method is used to notify the CASA on the limit change due to the amount schedule. This is called during the batch processing of the amount schedule. @param facilityNotificationDTO @throws FatalException	This method executes the FacilityAmountSchedule Processing taking arguments as appContext and batchData @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsNotificationAdapter	notifyOnScheduleChange	This method is used to notify the CASA on the limit change due to the amount schedule. This is called during the batch processing of the amount schedule. @param facilityNotificationDTO @throws FatalException	This method executes the FacilityAmountScheduleExpiry Processing taking arguments as appContext and batchData
IDDACustomerAdviceReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACustomerAdviceReportAdapter	deliverSIFailedAdvice		
IDDACustomerAdviceReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACustomerAdviceReportAdapter	deliverSIFailedAdvice		
IDDACustomerAdviceReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACustomerAdviceReportAdapter	deliverSIFailedAdvice		
IDDACustomerAdviceReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACustomerAdviceReportAdapter	deliverChequeBookIssueAdvice		
IDDACustomerAdviceReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACustomerAdviceReportAdapter	deliverAccountNonActivityReport		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProposedPayinAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProposedPayinAdapter	logProposedPayin	<p>This method is used to log the ProposedPayinDTO details for Term Deposit. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDProposedPayinDTO,proposedPayinDTO, ,This class has the minimal details captured for creating a proposed Term Deposit . This is processed when a payment is received towards the proposed term deposit. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProposedPayinAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProposedPayinAdapter	logProposedPayin	This method is used to log the ProposedPayinDTO details for Term Deposit. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDProposedPayinDTO,proposedPayinDTO, ,This class has the minimal details captured for creating a proposed Term Deposit . This is processed when a payment is received towards the proposed term deposit. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException	This method is used to log the ProposedPayinDTO details for Term Deposit. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDProposedPayinDTO,proposedPayinDTO, ,This class has the minimal details captured for creating a proposed Term Deposit . This is processed when a payment is received towards the proposed term deposit. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IDDAJournalEntryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAJournalEntryAdapter	createJournalBatch		
IMultiEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarketEntity	Loads the market entity code from the business unit code present in the session context @param sessionContext @return String market entity code @throws FatalException	Loads the market entity code from the business unit code present in the session context @return String market entity code @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarket Entity	Loads the market entity code from the business unit code present in the session context @param sessionContext @return String market entity code @throws FatalException	Loads the market entity name for the given market entity code @return String market entity code @throws FatalException
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarket Entity	Loads the market entity code from the business unit code present in the session context @param sessionContext @return String market entity code @throws FatalException	This method is used to fetch the name of the market entity name whose code we are passing. @param sessionContext @param marketEntityCode @return @throws FatalException
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarket Entity	Loads the market entity code from the business unit code present in the session context @param sessionContext @return String market entity code @throws FatalException	Common method to populate common business indicators for submission/application/productgroup linkage @param originationBusinessIndicatorDTO @param headerDTO
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	fetchMarket EntityDetails	Fetches the market entity code from the business unit code @param sessionContext @param targetUnitCode @return String market entity code @throws FatalException	Fetches the market entity code from the business unit code present in the session context. @param sessionContext @return String[] An array of following in the said order: market entity code, bank code, business unit @throws FatalException
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	fetchMarket EntityDetails	Fetches the market entity code from the business unit code @param sessionContext @param targetUnitCode @return String market entity code @throws FatalException	
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\me\IMultiEntityAdapter	fetchMarket EntityDetails	Fetches the market entity code from the business unit code @param targetUnitCode @return String market entity code @throws FatalException	Fetches the market entity code from the business unit code present in the session context. @param sessionContext @return String[] An array of following in the said order: market entity code, bank code, business unit @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	fetchMarketEntityDetails	Fetches the market entity code from the business unit code @param targetUnitCode @return String market entity code @throws FatalException	
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadBusinessUnit	Fetches and returns Business Unit Name from business unit code passed as an argument @param businessUnitCode @return	Fetches and returns Business Unit Name from business unit code passed as an argument @param businessUnitCode @return String Business Unit Name
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadBusinessUnit	Fetches and returns Business Unit Name from business unit code passed as an argument @param businessUnitCode @return	This method is used to fetch the name of the business unit name whose code we are passing. @param businessUnitCode @return @throws FatalException
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadBusinessUnit	Fetches and returns Business Unit Name from business unit code passed as an argument @param businessUnitCode @return	Common method to populate common business indicators for submission/application/productgroup linkage @param originationBusinessIndicatorDTO @param headerDTO
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	fetchRegulatoryRegionForBranch		
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarketEntityName	Fetches the market entity name from the market entity code @param marketEntityCode @return String market entity name @throws FatalException	Loads the market entity name for the given market entity code @return String market entity code @throws FatalException
IMultiEntity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarketEntityName	Fetches the market entity name from the market entity code @param marketEntityCode @return String market entity name @throws FatalException	This method is used to fetch the name of the market entity name whose code we are passing. @param sessionContext @param marketEntityCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMultiEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\me\IMultiEntityAdapter	loadMarketEntityName	Fetches the market entity name from the market entity code @param marketEntityCode @return String market entity name @throws FatalException	Common method to populate common business indicators for submission/application/productgroup linkage @param originationBusinessIndicatorDTO @param headerDTO
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	initialize		
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	fetchDataValue		This method fetches the data value associated with each input parameter. @fcb.param,MI,Strubg,parameterName, ,It represents the input parameter name. @return String It returns the data value associated with each input parameter.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDenormalizedEntityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\metadata\IDenormalizedEntityAdapter	fetchDataValue		This method takes the service inputs and extracts the data required for authorization by an asserter and returns the same as an array of name value pairs. The data is extracted either from the service inputs or a fixed set of entities that are configured and provided for out-of-the box for participating in the authorization process. The data that comes in as input for the service can be configured as directly available or indirectly available. Directly available input is one which would be a non-data object input and can be used directly in the authorization process. For example, transaction currency or transaction amount is taken as direct input in debit or credit services. An indirect input is a data element or attribute which can be obtained from a data object which come into the service as a direct input and hence needs to be extracted by recursing the object graph at runtime. An example of this would be the transaction branch input which is an attribute in the session context. In case where the data is taken from the domain entity, the key field data required to load an domain entity instance from the datastore is expected to be a direct or indirect input to the service call. @param inputs @return @throws FatalException
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	addBlackout	This method will add the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will add the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	updateBlackout	This method will update the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will update the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	deleteBlackout	This method will delete the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will delete the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	fetchBlackout	This method will fetch the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will fetch the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	reopenBlackout	This method will reopen the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will reopen the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	closeBlackout	This method will close the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.	This method will close the blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,TransactionBlackoutDTO,blackoutDTO, , Indicates the blackout DTO.
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	fetchUserBlackout	This method will fetch the user blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,String,role, , Indicates the role for which blackout to be fetched. @fcb.param MI,String,partyId, , Indicates the user.	This method will fetch the user blackout on a transaction . @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context. @fcb.param MI,String,role, , Indicates the role for which blackout to be fetched. @fcb.param MI,String,partyId, , Indicates the user.
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	checkBlackout	This method will checks for the user blackout and transaction blackout. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context	This method is used to test whether the transaction has been blocked for specific period or not. If the transaction has been performed within the blackout period, it will be blocked. @param sessionContext - SessionContext object @throws FatalException - Exception being returned
ITransactionBlackoutAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITransactionBlackoutAssertionAdapter	checkBlackout	This method will checks for the user blackout and transaction blackout. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context	This method will checks for the user blackout and transaction blackout. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , Indicates the session context

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSITransaction	This method will log the successfully executed standing instructions in the IB table. @param sessionContext @param executionStatus @param standingInstructionInformationDTO @throws FatalException	
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSITransaction	This method will log the successfully executed standing instructions in the IB table. @param sessionContext @param executionStatus @param standingInstructionInformationDTO @throws FatalException	
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSITransaction	This method will log the successfully executed standing instructions in the IB table. @param sessionContext @param executionStatus @param standingInstructionInformationDTO @throws FatalException	
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSITransaction	This method will log the successfully executed standing instructions in the IB table. @param sessionContext @param executionStatus @param standingInstructionInformationDTO @throws FatalException	This method will log the successfully executed standing instructions in the IB table. @param sessionContext @param executionStatus @param standingInstructionInformationDTO @throws FatalException
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSuccessfulFundsTransfer	This method will log the successfully executed fund transfer transactions into the IB table. @param fundsTransferTransactionDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACHannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACHannelAdapter	logSuccessfulFundsTransfer	<p>This method will log the successfully executed fund transfer transactions into the IB table.</p> <p>@param fundsTransferTransactionDTO</p> <p>@throws FatalException</p>	<p>This method is used to transfer funds from a given demand deposit account to a single or multiple beneficiary accounts as the case maybe. The method will be responsible for the following:</p> <ul style="list-style-type: none"> <li>1. Validate the business policies for a funds transfer transaction. If the policies are satisfied, debit the provider account and credit the beneficiary accounts.</li> <li>2. Calculate and levy the fees on the debit account.</li> <li>3. Raise accounting events.</li> <li>4. Validate the override conditions. If these override conditions are authorized by the authorizer based on the SMS setup, the transaction will be successful. The override conditions will be applicable for both debit and credit accounts.</li> </ul> <p>The funds transfer method works as follows:</p> <ul style="list-style-type: none"> <li>1. Call the assembler method <code>FundsTransferTransactionAssembler#toInformationObject</code> to convert the details of the incoming DTO to Information object <code>FundsTransferTransaction</code>.</li> <li>2. Validate if the transaction is duplicate by calling the service of the parent class <code>super.validateDuplicateTransaction</code>.</li> <li>3. Call the private method <code>checkFundsTransferBusinessPolicies</code> to validate the business rules for the transaction.</li> <li>4. Call the method <code>super.manageTransactionDates</code> for posting date validations.</li> <li>5. Get the transactionReferenceNumber from the sessionContext.</li> <li>6. Get the subsequence number using <code>TransactionAuditItemHelper#getNextSequenceNumber(transactionReferenceNumber: String): int</code>.</li> <li>7. Call <code>super.debitFunds(accountTransaction: AccountTransaction): DebitResponse</code> for debiting the provider account with the transaction amount.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> <ul style="list-style-type: none"> <li>&lt;li&gt;Call the private method creditBeneficiaryAccounts to credit the funds in the one or multiple beneficiary accounts as the case may be.&lt;/li&gt;</li> <li>&lt;li&gt;Call the private method {@link this.manageFee} to levy fees on the account . The fees will always be levied on the debit account.&lt;/li&gt;</li> <li>&lt;li&gt;Open a loop for the beneficiary accounts. &lt;/li&gt;</li> <li>&lt;li&gt;Create a new AccountTransaction for the credit account by copying the details of the debit AccountTransaction and updating the accountId and transactionAmounts with the beneficiary details.&lt;/li&gt;</li> <li>&lt;li&gt;Add the new AccountTransaction in the accountTransactionarray.&lt;/li&gt;</li> <li>&lt;li&gt;Log financial accounting event using {@link super#raiseFinancialEvents(accountTransacitonArray: AccountTransaction[], feeDetails: FeeDetails, instrumentId: String)}. Pass the accountTransactionarray created above to the service.&lt;/li&gt;</li> <li>TODO:// Amit to confirm whether a debit entry is to be passed for each credit entry on one debit entry should be passed with multiple credit entries. &lt;/li&gt;</li> <li>&lt;li&gt;Call the method {@link this.handleFundsTransferOverride} to check the override conditions.&lt;/li&gt;</li> </ul> <p> <ul style="list-style-type: none"> <li>@param sessionContext MI,SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</li> <li>@param fundsTransferTransactionDTO MI, FundTransferTransactionDTO, fundTransferTransactionDTO, This parameters stores the details of the funds transfer. It stores the details of the account and the amount to transfer. This DTO stores the information of the Beneficiary accounts and the amount that will be transfered. &lt;ol&gt;</li> <li>&lt;li&gt;FundsTransferType, MI, which specifies the type</li> </ul> </p> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>of type of transfer instruction, whether it is for single beneficiary account or multiple beneficiary accounts.</p> <p><b>BeneficiaryDetailsDTO[]</b>, MI. stores the information of the beneficiary accounts and the amount of transfer for each beneficiary. If the FundsTransferType is Single, at least one instance of BeneficiaryDetailsDTO is required. If FundsTransferType is multiple more than one instance of BeneficiaryDetailsDTO are required.</p> <p><b>@return</b> <a href="#">TransactionStatus</a>. The response contains the Transaction Status. The following sections indicate the completion of the transactions with reasons</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS</a></li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR</a></li> </ul> <p><b>@throws</b> <a href="#">com.ofss.fc.infra.exception.FatalException</a> Please refer to the documentation above for <a href="#">Transaction Failure</a> section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAChannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAChannelAdapter	logSuccessfulFundsTransfer	This method will log the successfully executed fund transfer transactions into the IB table. @param fundsTransferTransactionDTO @throws FatalException	This method is wrapper service for DashBoard to transfer funds,this method sets the response with hold amount,fees amount in case of provider accounts,sweep in amounts set on the beneficiary account.This method is used to transfer funds from a given demand deposit account to a single or multiple beneficiary accounts as the case maybe. The method will be responsible for the following:  <ol> <li>Validate the business policies for a funds transfer transaction. If the policies are satisfied, debit the provider account and credit the beneficiary accounts.</li> <li>Calculate and levy the fees on the debit account.</li> <li>Raise accounting events.</li> <li>Validate the override conditions. If these override conditions are authorized by the authorizer based on the SMS setup, the transaction will be successful. The override conditions will be applicable for both debit and credit accounts.</li> </ol> The funds transfer method works as follows:  <ul> <li>Call the assembler method {@link FundsTransferTransactionAssembler#toInformationObject} to convert the details of the incoming DTO to Information object FundsTransferTransaction.</li> <li>Validate if the transaction is duplicate by calling the service of the parent class {@link super.validateDuplicateTransaction}.</li> <li>Call the private method checkFundsTransferBusinessPolicies to validate the business rules for the transaction.</li> <li>Call the method {@link super.manageTransactionDates} for posting date validations.</li> <li>Get the transactionReferenceNumber from the sessionContext.</li> <li>Get the subsequence number using {@link TransactionAuditItemHelper#getNextSequenceNumber}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>er(transactionReferenceNumber: String): int}./li&gt;  &lt;li&gt;Call {@link super.debitFunds(accountTransaction: AccountTransaction): DebitResponse} for debiting the provider account with the transaction amount./li&gt;  &lt;li&gt;Call the private method creditBeneficiaryAccounts to credit the funds in the one or multiple beneficiary accounts as the case may be./li&gt; &lt;li&gt;Call the private method {@link this.manageFee} to levy fees on the account . The fees will always be levied on the debit account./li&gt; &lt;li&gt;Open a loop for the beneficiary accounts. &lt;li&gt;Create a new AccountTransaction for the credit account by copying the details of the debit AccountTransaction and updating the accountId and transactionAmounts with the beneficiary details./li&gt;  &lt;li&gt;Add the new AccountTransaction in the accountTransactionarray./li&gt; &lt;li&gt;Log financial accounting event using {@link super#raiseFinancialEvents(accountTransacitonArray: AccountTransaction[], feeDetails: FeeDetails, instrumentId: String)}. Pass the accountTransactionarray created above to the service./li&gt; TODO:// Amit to confirm whether a debit entry is to be passed for each credit entry on one debit entry should be passed with multiple credit entries. &lt;li&gt;Call the method {@link this.handleFundsTransferOverride} to check the override conditions./li&gt; &lt;/ul&gt; @param sessionContext MI,SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param fundsTransferTransactionDTO MI, FundTransferTransactionDTO, fundTransferTransactionDTO, This parameters stores the details of the funds transfer. It stores the details of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the account and the amount to transfer. This DTO stores the information of the Beneficiary accounts and the amount that will be transferred. &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;FundsTransferType, MI, which specifies the type of type of transfer instruction, whether it is for single beneficiary account or multiple beneficiary accounts.&lt;/li&gt;</li> <li>&lt;li&gt;BeneficiaryDetailsDTO[], MI. stores the information of the beneficiary accounts and the amount of transfer for each beneficiary. If the FundsTransferType is Single, at least one instance of BeneficiaryDetailsDTO is required. If FundsTransferType is multiple more than one instance of BeneficiaryDetailsDTO are required.&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;</p> <p>@return {@link TransactionStatus}. The response contains the Transaction Status. The following sections indicate the completion of the transactions with reasons &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;b&gt;Transaction Success&lt;/b&gt;</li> <li>&lt;ul&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;</li> <li>&lt;/ul&gt;</li> <li>&lt;b&gt;Transaction Failure&lt;/b&gt;</li> <li>&lt;ul&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt;</li> <li>&lt;/ul&gt;</li> </ul> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	<p>This method will fetch the latest rule definition for the rule id by interacting with rules module.</p> <p>@param sessionContext @param ruleId @return RuleDefinition @throws FatalException</p>	<p>This method is used to set rule definition in Condition policy dto. @param sessionContext @param domainObjectDTOs @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	This method is used to set rule definition in Tax Parameter dto. @param sessionContext @param domainObjectDTOs @throws FatalException
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	This method is used to set rule definition in Condition policy dto. @param sessionContext @param domainObjectDTOs @throws FatalException
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	This method is used to set rule definition in Covenant policy dto. @param sessionContext @param domainObjectDTOs @throws FatalException
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	@see com.ofss.fc.app.adapter.integration.fsm.IFSMRuleAdapter#viewLatestRuleForProcessDate(com.ofss.fc.app.context.SessionContext, java.lang.String)
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\fsm\IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	This method is used to set various fields in LayoutDetailsPolicy, which are not set on hibernate fetch. @param sessionContext @param domainObjectDTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\ fsm \IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	This method is used to set rule definition in interest rule dto. @param sessionContext @param domainObjectDTOs @throws FatalException
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\ fsm \IFSMRuleAdapter	viewLatestRuleForProcessDate	This method will fetch the latest rule definition for the rule id by interacting with rules module. @param sessionContext @param ruleId @return RuleDefinition @throws FatalException	@param sessionContext @param interestRuleDTO @return @throws FatalException
IFSMRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\ fsm \IFSMRuleAdapter	viewRuleForProcessDate	This method will fetch the rule definition for the process date and rule id by interacting with rules module. @param sessionContext @param ruleId @param effectiveDate @return @throws FatalException	This method is used to set rule definition in Covenant policy dto. @param sessionContext @param domainObjectDTOs @throws FatalException
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This	This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			string contains the pricing ticket number returned by pricing engine	application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception   {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	<p>@fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI,TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket	This method populates pricing context without setting the facts. @param applicationContext @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			number returned by pricing engine	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This service is used for creating Pricing Plan in pricing module.   Pricing plan in Origination is created for following scenarios:   1) For lending product groups, origination fees are applicable and can   be collected even before the submission is submitted.  Facility Id is used as the arrangement number for collecting such fees.  2) In case of bundle applied, bundle level subscription fees are   applicable and be collected.  Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This service will fetch origination fees for lending product group being applied by the applicant. The following activities will be carried out in sequence.  1. Pricing Plan will be created using combination of submission and facility Id as the arrangement number.  2. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  3. This ticket number is stored in Origination to refresh fees related to the event.   4. Pricing module is called to get pricing information for this New Facility or Amend   Facility Event.  @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant information. @fcb.param MI, CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details. @fcb.param MI, CollateralDTO[], collateralDTOs, , Indicates the collateral details. @return {@link PricingResponse} This class holds ticket number for origination fees which is stored.  All inquiries for event associated with facility id is made using this ticket number.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This service is used for creating pricing plan for bundle subscription fees.  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for this pricing plan.  This reference Id is generated using the account module and is generated only once   for the given submission. For all product groups under the submission, the same bundle  reference Id is to be used.  2. The Pricing Plan is created using this bundle reference Id.   3. A new pricing ticket number is generated and stored in Origination for Bundle Fees.  4. Pricing Module is called to get pricing information for this Bundle Subscription Event.  Fees can be collected, waived or deferred for this bundle reference Id.   The pricing ticket number stored for bundle fees is used for tracking such   fees collected against this arrangement number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.  All inquiries for this bundle reference Id are made using this ticket number.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further	This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine</p>	<p>this pricing plan.&lt;br&gt; This reference Id is generated using the account module and is generated only once &lt;br&gt; for the given submission.&lt;br&gt; 2. The Pricing Plan is created using this bundle reference Id. &lt;br&gt; 3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.&lt;br&gt; 4. Pricing Module is called to get pricing information for this Bundle Subscription Event.&lt;br&gt; 5. If bundle subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.&lt;br&gt; 6. Else Fees can be negotiated, waived or deferred for this bundle reference Id. &lt;br&gt; The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO, ,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.&lt;br&gt; All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This service will fetch the fees applied for withdrawing the submission.  The following activities will be carried out in sequence.  1. Pricing Ticket Number is generated and is used for creating price recommendations.  2. Price Recommendations will be created for the submissionId.  3. Pricing Plan will be created using the above submissionId as the arrangement number.  4. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  5. This ticket number is stored in Origination to refresh fees related to the event.   6. Pricing module is called to get pricing information for this OR_SUBMISSION_WITHDRAW Event.  @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the identifier of the submission @return {@link PricingResponse} This class holds ticket number for withdrawal fees which is stored. 
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context	This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine</p>	<p>which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	<p>This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates,margins and servicing fees are done using the</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration.Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine</p>	<p>same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	This service will confirm structure solution for lending product group in a submission.   Various operations performed during this service are as follows:   <ol> <li>The facility structure is validated as per the guidelines. {@link LendingApplicationHelper#checkFacilityConformity}</li>  <li>Validations for confirming structure solution are validated. {@link LendingApplicationHelper#validateConfirmActionOnStructureSolution}</li>  <li>Check if ticket number for Origination fees is present. If yes, defer all the uncollected fees for that ticket number else create a new pricing plan, fetch and defer the origination fees.</li>  <li>Check if product group is setup under a bundle and ticket number for bundle fees is present. If yes, defer all the fees for that ticket number else create a new pricing plan, fetch and defer the bundle fees.</li>  <li>In case of Amended facility, check if settlement is required or not (irrespective of the settlement flag on product group and update the same) {@link LendingApplicationHelper#checkIfSettlementRequiredForPG}</li>  <li>For Loan product groups, update party id, borrowing entity and facility details for loan simulations</li>  <li>Check for any rates and fee negotiations. If present, send for discretionary pricing approval</li>  </ol> @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @return LendingAccountCreationResponse, this class returns

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the LendingAccountCreationResponse object,  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#KYC_DETAILS_NOT_FO  UND} : FC_OR_PI_0001 &lt;br&gt; This check is part of  Pre-create account validation. &lt;br&gt; This exception is  thrown if KYC has not been confirmed for all parties  &lt;br&gt; applying to the product group for which account  is being opened. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_DPA_APPROVAL} :  FC_OR_DE_0082 &lt;br&gt; This exception is raised as an  overridable exception when pricing rates and fees are  negotiated and require approval. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_AMOUNT_  GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception  occurs when one or more sub facilities are configured  and the requested amount for these sub-lines exceed  their parent facility's (can be main facility or another  sub facility) amount. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_TERM_GRE  ATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs  when one or more sub facilities are configured and the  requested term for these sub-lines exceed their parent  facility's (can be main facility or another sub facility)  term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_  SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#LVR_THRESHOLD_BREA  CHED} : FC_OR_SS_0006 &lt;br&gt; This exception occurs  when the loan to value ratio is breached beyond the  limit allowed for a product group. This threshold  value is maintained in PMU in credit policy template</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>for the product group &lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER</a> : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the a single facility or sub-facility is greater than requested amount of that facility. &lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER_THAN_MAIN_FACILITY</a> : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the facility structure exceed the total amount of main facility. &lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#ACCT_NOT_CONFIGURED</a> : FC_OR_SS_0015 &lt;br&gt; This exception occurs when structure solution is being confirmed with no account configured under the facility. &lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#FETCH_PRICE_OFFERINGS_FAILED</a> : FC_OR_SS_0016 &lt;br&gt; This exception occurs when fetching of price recommendations for origination fees from pricing module fails.&lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#PPC_CREATION_FAILED</a> : FC_OR_SS_0017 &lt;br&gt; This exception occurs when creation of pricing plan for origination fees in pricing module fails.&lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#GET_PRICE_ON_EVENT_FAILED</a> : FC_OR_SS_0018 &lt;br&gt; This exception occurs when fetching price for events in origination (OR_NEW_FACILITY or OR_AMEND_FACILITY) from pricing module fails.&lt;br&gt; &lt;br&gt; <a href="#">@link OriginationErrorConstants#OD_FIXED_TERM_GREATER</a> : FC_OR_SS_0019 &lt;br&gt; This exception occurs when fixed term entered for overdraft product groups</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>exceed the parent requested limit term.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#APPLICANTS_HOLDER_PREFERENCES_INCORRECT} : FC_OR_SS_0021&lt;br&gt; This exception occurs when user configures the account configuration in structure solution and then goes back to the product details page and changes the applicants applying for the product group.&lt;br&gt; The user is prompted to check the account configuration again to change the account holder preference for changed applicants.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_FEE_NEGOTIATION} : FC_PR_FEE_OVR &lt;br&gt; This exception occurs any of the rates or fees are negotiated.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#COLLATERAL_VALUATION_INCOMPLETE} : FC_OR_CL_0001 &lt;br&gt; This exception occurs if NCCP is yes and valuation for any collaterals linked to the facility is not yet complete.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	<ul style="list-style-type: none"> <li>This method populates the following details for a given lending product group linkage. </li> <li>1. Populates the necessary facts requires for the Pricing call. </li> <li>2. Generates the pricing ticket number by initiating the start dialogue method. </li> <li>3. Fetches the list of pricing fund items applicable for the given funding template id's. </li> <li>4. Make the pricing call with all necessary facts and fetches the pricing fees for the product group. </li> <li>5. Updates the pricing ticket number in the product group linkage. </li> <li>6. Checks whether the LMI is applicable for the product group. If applicable it populates the LMI funding details. </li> <li>7. Populate CCI amount </li> </ul> LMI should be fetched after CapitalizationDetailsDTO being populated so that it will take into acc the capitalization amount @param sessionContext @param submissionInputDTO @param lendingPgLinkageDTO @param creditFacilityDTOs @return {@link FundingTableDetailDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	startDialogue	This service is used for initiating a dialogue with the pricing module. The pricing module returns a unique pricing ticket number which is fetched for each account configuration. All further interactions pertaining to that account configuration such as fetching interest rates, margins and servicing fees are done using the same ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @return String, , This string contains the pricing ticket number returned by pricing engine	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchInterestRatesForDDA	This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the interest rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO,, Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchInterestRatesForDDA	<p>This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the interest rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber, , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO,, Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
<p>IOrigination PricingAdapter</p>	<p>com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter</p>	<p>fetchInterestRatesForDDA</p>	<p>This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the interest rates</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO,, Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchInterestRatesForTerm	<p>This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the interest rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOrginationPricingAdapter	fetchInterestRatesForTerm	<p>This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date,</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate, , This is the date with respect to which the interest rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>product group details for the account configuration @fcb.param MI, ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchInterestRatesForTD	<p>This service is used for fetching the interest rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the interest rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchMarginsForDDA	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date,</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate, , This is the date with respect to which the margin rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchMarginsForDDA	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the margin rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOriginatio PricingAdapt er	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nPricingAdapter	fetchMargi nsForDDA	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of interest rates. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the margin rates</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchMarginsForTD	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of margin rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the margin rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
<p>IOrigination PricingAdapter</p>	<p>com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter</p>	<p>fetchMarginsForTD</p>	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of margin rates. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date,</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate, , This is the date with respect to which the margin rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	<p>product group details for the account configuration @fcb.param MI, ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchMarginsForTD	<p>This service is used for fetching the margin rates for the account configuration from the pricing engine. The rates that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of margin rates.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the margin rates need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation</p>	
IOriginat PricingAdapt er	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nPricingAdapter	fetchServi ngFeesForD DA	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date,</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, DDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchServicingFeesForDDA	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, DDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchServicingFeesForDDA	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a demand deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, DDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchServicingFeesForTD	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing that is the interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return String, , This string contains the pricing ticket number. This ticket number will be used to inquire the pricing details setup for the account configuration for display from pricing engine. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, TDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchServicingFeesForTD	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, TDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	<p>occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchServicingFeesForTD	<p>This service is used for fetching the servicing charges for the account configuration from the pricing engine. The charges that are fetched are with respect to the effective date that is passed to pricing engine. All interactions for an account configuration are done using the pricing ticket number which is returned by pricing when a dialogue for the account configuration is initiated. Finally, all facts for the application are required for calculation of servicing charges.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, factsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @fcb.param MI, TDAccountConfigurationDTO, accountConfigurationDTO, , Indicates the account parameters which are configured during the structure solution and for which servicing fees are being fetched</p>	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getNetInterestRate	<p>This service is used for fetching the net interest rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , ,</p>	<p>This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, ddaFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the net interest rate for the product @throws FatalException</p>	<p>information such as pricing ticket number unique for each account configuration and net interest rate.</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getNetInterestRate	<p>This service is used for fetching the net interest rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date,</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, ddaFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the net interest rate for the product @throws FatalException</p>	<p>the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getNetInterestRate	<p>This service is used for fetching the net interest rate for facts being passed for the product and party.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, ddaFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information,</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the net interest rate for the product @throws FatalException	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getNetInterestRate	This service is used for fetching the net interest rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, Map<String, Object>, ddaFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the net interest rate for the product @throws FatalException</p>	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getNetInterestRate	<p>This service is used for fetching the net interest rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>identifier for the product group under which the account is being setup @fcb.param MI, Map&lt;String, Object&gt;, ddaFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the net interest rate for the product @throws FatalException</p>	<p>is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOriginatio PricingAdapt er	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \origination\IOriginatio nPricingAdapter	getInterestS ensitivityRa te	<p>This service is used for fetching the interest sensitivity rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param</p>	<p>This service is used for fetching the interest sensitivity rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, productOfferingId, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, OriginationFactsDTO, originationFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, OriginationFactsDTO, originationFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the interest sensitivity rate for the product @throws FatalException</p>	<p>servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the interest sensitivity rate for the product @throws FatalException</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getInterestSensitivityRate	<p>This service is used for fetching the interest sensitivity rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, OriginationFactsDTO, originationFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the interest sensitivity rate for the product @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getInterestSensitivityRate	<p>This service is used for fetching the interest sensitivity rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, OriginationFactsDTO, originationFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor,</p>	<p>This method calculates the automatic Debt Servicing Obligation (DSO) amounts for each facility decision to be used at UI as part of repayment summary. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionDecisionDTO, submissionDecisionDTO, , {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionDTO} This object holds all the details of decision for a given submission @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the interest sensitivity rate for the product @throws FatalException</p>	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getInterestSensitivityRate	<p>This service is used for fetching the interest sensitivity rate for facts being passed for the product and party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, Date, effectiveDate, , This is the date with respect to which the servicing charges need to be fetched for the account configuration @fcb.param MI, String, pricingTicketNumber , , This string denotes the pricing ticket number fetched from pricing engine when dialogue is initiated for the account configuration @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the account is being setup @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group under which the account is being setup @fcb.param MI, OriginationFactsDTO,</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>originationFactsDTO, , Indicates the facts that are passed for a term deposit account configuration for fetching pricing related information such as interest rates, margins and any servicing fees is applicable. Various facts such as requested amount, requested tenor, applicant(s) related information, product related information, etc. are passed to Pricing module for evaluation @return BigDecimal, , Indicates the interest sensitivity rate for the product @throws FatalException</p>	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchOriginationFees	<p>This service is used for fetching the Origination fees for facts being passed for the applicant and product group. @param fundingTableDetailDTO @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, FundingFactsDTO[], fundingFactsDTOArray, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException</p>	<p>This method is used to fetch origination fees using the application id and facility id. The logic of the method is as follows&lt;br&gt; &lt;ol&gt; &lt;li&gt;Fetch pricing ticket number using the service @link{OriginationFeeApplicationService#fetchOriginationFees(SessionContext, HeaderDTO, LendingProductGroupLinkageDTO, ApplicantDataDTO, CreditFacilityDTO[], CollateralDTO[])}.&lt;/br&gt; The input of This service can be derived as follows,&lt;br&gt; &lt;ul&gt; &lt;li&gt;SessionContext - SessionContext from the input HeaderDTO - Inquire application details using base application service @link{com.ofss.fc.app.origination.service.core.application.BaseApplicationService.inquireApplication} using the application id, input. Pass the header dto from the ApplicationDTO retrieved.&lt;/li&gt; &lt;li&gt;LendingProductGroupLinkageDTO - Fetch product group linkage from the ApplicationDTO retrieved above, using the facility id, input. Set the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingTicketNumberForBundleFees to the pg linkage dto, if IsNominatedForBundle() and PricingTicketNumberForBundleFees() is empty, by getting the new pricing ticket number by invoking the service</p> <ul style="list-style-type: none"> <li>@link{OriginationFeeApplicationService#createBundleSubscriptionPricingPlan &lt;/li&gt; &lt;li&gt;ApplicantDataDTO &lt;li&gt;Fetch all applicants for the application using the helper method</li></ul> <ul style="list-style-type: none"> <li>@link{LendingApplicationHelper#fetchAllApplicantsForApplicationId &lt;/li&gt; &lt;li&gt;Convert applicant DTO To applicant data DTO using the helper method</li> <li>@link{SubmissionInquiryHelper#populateApplicantDataDTO &lt;/li&gt;&lt;/li&gt; &lt;li&gt;CreditFacilityDTO[] - Fetch credit facility array DTO using the service</li> <li>@link{LendingSimulationApplicationService#fetchFacilityHierarchy &lt;/li&gt; &lt;li&gt;CollateralDTO[] - Fetch collateral using the service</li> <li>@link{LendingApplicationService#fetchCollateralsForApplication}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;</li> <li>&lt;li&gt;Use the pricing ticket number and get the pricing details using the service</li> <li>@link{OriginationPricingAdapter#fetchFeesForTicketNumber . The input facts can be populated using the methods</li> <li>@link{LendingApplicationService#populateFactsForOriginationFee) and</li> <li>@link{LendingApplicationService#populateFundingFactDefaultValues &lt;/li&gt; &lt;li&gt;Iterate WrappedPriceFeatureInquiryResponse #wrappedPriceFeatureDTO[], convert the WrappedPriceFeatureDTO to PricingFeeDetailDTO using the builder</li> <li>@link{PricingFeeDetailDTOBuilder#build &lt;/li&gt;</li> <li>&lt;li&gt;Set the pricing ticket number and PricingFeeDetail</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>DTOs in the response&lt;/li&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the origination fees is to be retrieved @fcb.param MI,String,facilityId, ,This object holds the facility id, for which the origination fees is to to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.pricing.PricingFeeResponse}, This contains the information on origination fees (pricingTicketNumber and pricingFeeDetails) @throws com.ofss.fc.infra.exception.FatalException</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchOriginationFees	<p>This service is used for fetching the Origination fees for facts being passed for the applicant and product group. @param fundingTableDetailDTO @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, FundingFactsDTO[], fundingFactsDTOArray, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException</p>	<p>This service is used for fetching the pricing fees for facts being passed for the applicant and product group. It will create a new pricing plan for the given arrangement number(FacilityId). After creating the pricing plan it will get the pricing fees by making call to the service getPriceOnEvent. After getting the price information it will populate the respective fundingItems and store it in the fundingTableDetail. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupId, , This is the identifier of the product group @fcb.param MI, Date, effectiveDate, , This is the effective Date @fcb.param MI, String, pricingTicketNumber, , This is the array facts which are used to calculate the fees @fcb.param MI, String, facilityId, , This is the array facts which are used to calculate the fees @fcb.param MI, String, eventCode, , This is the array facts which are used to calculate the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				fees @fcb.param MI, Map<String, Object>, fundingTableFacts, , This is the array facts which are used to calculate the fees @fcb.param MI, FundingTableDetailDTO, fundingTableDetailDTO, , This is the array facts which are used to calculate the fees @fcb.param MI, List<String>, pricngPurposeList, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchOriginationFees	This service is used for fetching the Origination fees for facts being passed for the applicant and product group. @param fundingTableDetailDTO @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, FundingFactsDTO[], fundingFactsDTOArray, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException	This service will fetch origination fees for lending product group being applied by the applicant.  The following activities will be carried out in sequence.  1. Pricing Plan will be created using combination of submission and facility Id as the arrangement number.  2. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  3. This ticket number is stored in Origination to refresh fees related to the event.   4. Pricing module is called to get pricing information for this New Facility or Amend   Facility Event.  @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant information. @fcb.param MI, CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details. @fcb.param MI, CollateralDTO[], collateralDTOs, , Indicates the collateral details. @return {@link PricingResponse} This class holds ticket number for origination fees which is stored.  All inquiries for event associated with facility id is made using this

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ticket number.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchOriginationFees	This service is used for fetching the Origination fees for facts being passed for the applicant and product group. @param fundingTableDetailDTO @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, FundingFactsDTO[], fundingFactsDTOArray, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException	<ul>This method populates the following details for a given lending product group linkage. </ul> <ul> <ul> 1. Populates the necessary facts requires for the Pricing call. </ul> <ul> 2. Generates the pricing ticket number by initiating the start dialogue method. </ul> <ul> 3. Fetches the list of pricing fund items applicable for the given funding template id's. </ul> <ul> 4. Make the pricing call with all necessary facts and fetches the pricing fees for the product group. </ul> <ul> 5. Updates the pricing ticket number in the product group linkage. </ul> <ul> 6. Checks whether the LMI is applicable for the product group. If applicable it populates the LMI funding details. </ul> <ul> 7. Populate CCI amount </ul> LMI should be fetched after CapitalizationDetailsDTO being populated so that it will take into acc the capitalization amount @param sessionContext @param submissionInputDTO @param lendingPgLinkageDTO @param creditFacilityDTOs @return {@link FundingTableDetailDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchOriginationFees	This service is used for fetching the Origination fees for facts being passed for the applicant and product group. @param fundingTableDetailDTO @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, FundingFactsDTO[], fundingFactsDTOArray, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException	
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPricingEventContextDTO	This method is called to create the PricingEventContextDTO from the factMap. @param sessionContext @param facts @param pricingTicket @param domainCategory @return @throws FatalException	This method will save the discounted amount for the PD @param sessionContext @param pricingTicketNumber @param priceCode @param discountedPriceDTO @param currencyCodes @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPricingEventContextDTO	This method is called to create the PricingEventContextDTO from the factMap. @param sessionContext @param facts @param pricingTicket @param domainCategory @return @throws FatalException	This service will populate all product group, facility, collateral and party facts for the given product group linkage.   1. All product group facts are derived from the Product Group linkage dto passed.  2. For the PG, facility Id is derived. A call to LCM is made to retrieve the facility details.   All facility facts are populated for the main facility.  3. Facility Collateral Linkages are derived from the above inquiry and a fetch on the collateral  details is done. Collateral facts for each collateral are then populated.  4. For the PG, all applicants linked to the PG are retrieved using the ApplicantProductGroupReference.  For each applicant, details from party are fetched and party facts are populated. @param sessionContext @param lendingProductGroupLinkageDTO @param applicantDataDTO @param creditFacilityDTOs @param collateralDTOs @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPricingEventContextDTO	This method is called to create the PricingEventContextDTO from the factMap. @param sessionContext @param facts @param pricingTicket @param domainCategory @return @throws FatalException	This method will fetch all facts applicable for the submissionId. @param sessionContext @param sessionContext @return OriginationFactResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	This service is used for creating Pricing Plan in pricing module.   Pricing plan in Origination is created for following scenarios:   1) For lending product groups, origination fees are applicable and can   be collected even before the submission is submitted.  Facility Id is used as the arrangement number for collecting such fees.  2) In case of bundle applied, bundle level subscription fees are   applicable and be collected.  Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException	This service is used for fetching the pricing fees for facts being passed for the applicant and product group. It will create a new pricing plan for the given arrangement number(FacilityId). After creating the pricing plan it will get the pricing fees by making call to the service getPriceOnEvent. After getting the price information it will populate the respective fundingItems and store it in the fundingTableDetail. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupId, , This is the identifier of the product group @fcb.param MI, Date, effectiveDate, , This is the effective Date @fcb.param MI, String, pricingTicketNumber, , This is the array facts which are used to calculate the fees @fcb.param MI, String, facilityId, , This is the array facts which are used to calculate the fees @fcb.param MI, String, eventCode, , This is the array facts which are used to calculate the fees @fcb.param MI, Map<String, Object>, fundingTableFacts, , This is the array facts which are used to calculate the fees @fcb.param MI, FundingTableDetailDTO, fundingTableDetailDTO, , This is the array facts which are used to calculate the fees @fcb.param MI, List<String>, pricngPurposeList, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	This service is used for creating Pricing Plan in pricing module.   Pricing plan in Origination is created for following scenarios:   1) For lending product groups, origination fees are applicable and can   be collected even before the submission is submitted.  Facility Id is used as the arrangement number for collecting such fees.  2) In case of bundle applied, bundle level subscription fees are   applicable and be collected.  Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException	This service will fetch origination fees for lending product group being applied by the applicant.  The following activities will be carried out in sequence.  1. Pricing Plan will be created using combination of submission and facility Id as the arrangement number.  2. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  3. This ticket number is stored in Origination to refresh fees related to the event.   4. Pricing module is called to get pricing information for this New Facility or Amend   Facility Event.  @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant information. @fcb.param MI, CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details. @fcb.param MI, CollateralDTO[], collateralDTOs, , Indicates the collateral details. @return {@link PricingResponse} This class holds ticket number for origination fees which is stored.  All inquiries for event associated with facility id is made using this ticket number.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	This service is used for creating Pricing Plan in pricing module.   Pricing plan in Origination is created for following scenarios:   1) For lending product groups, origination fees are applicable and can   be collected even before the submission is submitted.  Facility Id is used as the arrangement number for collecting such fees.  2) In case of bundle applied, bundle level subscription fees are   applicable and be collected.  Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException	This service is used for creating pricing plan for bundle subscription fees.  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for this pricing plan.  This reference Id is generated using the account module and is generated only once   for the given submission. For all product groups under the submission, the same bundle  reference Id is to be used.  2. The Pricing Plan is created using this bundle reference Id.   3. A new pricing ticket number is generated and stored in Origination for Bundle Fees.  4. Pricing Module is called to get pricing information for this Bundle Subscription Event.  Fees can be collected, waived or deferred for this bundle reference Id.   The pricing ticket number stored for bundle fees is used for tracking such   fees collected against this arrangement number. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkageDTO, Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.  All inquiries for this bundle reference Id are made using this ticket number.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	This service is used for creating Pricing Plan in pricing module.   Pricing plan in Origination is created for following scenarios:   1) For lending product groups, origination fees are applicable and	This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>can &lt;br&gt; be collected even before the submission is submitted.&lt;br&gt; Facility Id is used as the arrangement number for collecting such fees.&lt;br&gt; 2) In case of bundle applied, bundle level subscription fees are &lt;br&gt; applicable and be collected.&lt;br&gt; Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException</p>	<p>this pricing plan.&lt;br&gt; This reference Id is generated using the account module and is generated only once &lt;br&gt; for the given submission.&lt;br&gt; 2. The Pricing Plan is created using this bundle reference Id. &lt;br&gt; 3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.&lt;br&gt; 4. Pricing Module is called to get pricing information for this Bundle Subscription Event.&lt;br&gt; 5. If bundle subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.&lt;br&gt; 6. Else Fees can be negotiated, waived or deferred for this bundle reference Id. &lt;br&gt; The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO, ,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.&lt;br&gt; All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	<p>This service is used for creating Pricing Plan in pricing module. &lt;br&gt; Pricing plan in Origination is created for following scenarios: &lt;br&gt; 1) For lending product groups, origination fees are applicable and can &lt;br&gt; be collected even before the submission is submitted.&lt;br&gt; Facility Id is used as the arrangement number for collecting such fees.&lt;br&gt; 2) In case of bundle applied, bundle level subscription fees are &lt;br&gt; applicable and be collected.&lt;br&gt; Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException</p>	<p>This service will fetch the fees applied for withdrawing the submission.&lt;br&gt; The following activities will be carried out in sequence.&lt;br&gt; 1. Pricing Ticket Number is generated and is used for creating price recommendations.&lt;br&gt; 2. Price Recommendations will be created for the submissionId.&lt;br&gt; 3. Pricing Plan will be created using the above submissionId as the arrangement number.&lt;br&gt; 4. A new pricing ticket number is now generated and associated with the event for which&lt;br&gt; price is to be fetched.&lt;br&gt; 5. This ticket number is stored in Origination to refresh fees related to the event. &lt;br&gt; 6. Pricing module is called to get pricing information for this OR_SUBMISSION_WITHDRAW Event.&lt;br&gt; @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the identifier of the submission @return {@link PricingResponse} This class holds ticket number for withdrawal fees which is stored.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	createPricingPlan	<p>This service is used for creating Pricing Plan in pricing module. &lt;br&gt; Pricing plan in Origination is created for following scenarios: &lt;br&gt;</p> <ol style="list-style-type: none"> <li>1) For lending product groups, origination fees are applicable and can &lt;br&gt; be collected even before the submission is submitted.&lt;br&gt; Facility Id is used as the arrangement number for collecting such fees.&lt;br&gt;</li> <li>2) In case of bundle applied, bundle level subscription fees are &lt;br&gt; applicable and be collected.&lt;br&gt; Bundle Reference Id is used as the arrangement number for collecting such fees. @param sessionContext @param effectiveDate @param productGroupId @param offerId @param bundleId @param arrangementNumber @param facts @throws FatalException</li> </ol>	<p>This service will confirm structure solution for lending product group in a submission. &lt;br&gt; Various operations performed during this service are as follows: &lt;br&gt;</p> <ol style="list-style-type: none"> <li>&lt;li&gt;The facility structure is validated as per the guidelines. {@link LendingApplicationHelper#checkFacilityConformity}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Validations for confirming structure solution are validated. {@link LendingApplicationHelper#validateConfirmActionOnStructureSolution}&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if ticket number for Origination fees is present. If yes, defer all the uncollected fees for that ticket number else create a new pricing plan, fetch and defer the origination fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check if product group is setup under a bundle and ticket number for bundle fees is present. If yes, defer all the fees for that ticket number else create a new pricing plan, fetch and defer the bundle fees.&lt;/li&gt;&lt;br&gt; &lt;li&gt;In case of Amended facility, check if settlement is required or not (irrespective of the settlement flag on product group and update the same) {@link LendingApplicationHelper#checkIfSettlementRequiredForPG}&lt;/li&gt;&lt;br&gt; &lt;li&gt;For Loan product groups, update party id, borrowing entity and facility details for loan simulations&lt;/li&gt;&lt;br&gt; &lt;li&gt;Check for any rates and fee negotiations. If present, send for discretionary pricing approval&lt;/li&gt;&lt;br&gt; &lt;/ol&gt; @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @return LendingAccountCreationResponse, this class returns</li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the LendingAccountCreationResponse object,  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#KYC_DETAILS_NOT_FOUND} : FC_OR_PI_0001 &lt;br&gt; This check is part of  Pre-create account validation. &lt;br&gt; This exception is  thrown if KYC has not been confirmed for all parties  &lt;br&gt; applying to the product group for which account  is being opened. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_DPA_APPROVAL} :  FC_OR_DE_0082 &lt;br&gt; This exception is raised as an  overridable exception when pricing rates and fees are  negotiated and require approval. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_AMOUNT_&lt;br&gt; GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception  occurs when one or more sub facilities are configured  and the requested amount for these sub-lines exceed  their parent facility's (can be main facility or another  sub facility) amount. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_TERM_GRE&lt;br&gt; ATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs  when one or more sub facilities are configured and the  requested term for these sub-lines exceed their parent  facility's (can be main facility or another sub facility)  term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_&lt;br&gt; SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#LVR_THRESHOLD_BREA&lt;br&gt; CHED} : FC_OR_SS_0006 &lt;br&gt; This exception occurs  when the loan to value ratio is breached beyond the  limit allowed for a product group. This threshold  value is maintained in PMU in credit policy template</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>for the product group &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the a single facility or sub-facility is greater than requested amount of that facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER_THAN_MAIN_FACILITY} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the facility structure exceed the total amount of main facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#ACCT_NOT_CONFIGURED} : FC_OR_SS_0015 &lt;br&gt; This exception occurs when structure solution is being confirmed with no account configured under the facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FETCH_PRICE_OFFERINGS_FAILED} : FC_OR_SS_0016 &lt;br&gt; This exception occurs when fetching of price recommendations for origination fees from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PPC_CREATION_FAILED} : FC_OR_SS_0017 &lt;br&gt; This exception occurs when creation of pricing plan for origination fees in pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#GET_PRICE_ON_EVENT_FAILED} : FC_OR_SS_0018 &lt;br&gt; This exception occurs when fetching price for events in origination (OR_NEW_FACILITY or OR_AMEND_FACILITY) from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OD_FIXED_TERM_GREATER} : FC_OR_SS_0019 &lt;br&gt; This exception occurs when fixed term entered for overdraft product groups</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>exceed the parent requested limit term.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#APPLICANTS_HOLDER_PREFERENCES_INCORRECT} : FC_OR_SS_0021&lt;br&gt; This exception occurs when user configures the account configuration in structure solution and then goes back to the product details page and changes the applicants applying for the product group.&lt;br&gt; The user is prompted to check the account configuration again to change the account holder preference for changed applicants.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_FEE_NEGOTIATION} : FC_PR_FEE_OVR &lt;br&gt; This exception occurs any of the rates or fees are negotiated.&lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#COLLATERAL_VALUATION_INCOMPLETE} : FC_OR_CL_0001 &lt;br&gt; This exception occurs if NCCP is yes and valuation for any collaterals linked to the facility is not yet complete.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service is used for fetching the pricing fees for facts being passed for the applicant and product group. It will create a new pricing plan for the given arrangement number(FacilityId). After creating the pricing plan it will get the pricing fees by making call to the service getPriceOnEvent. After getting the price information it will populate the respective fundingItems and store it in the fundingTableDetail. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupId, , This is the identifier of the product group @fcb.param MI, Date, effectiveDate, , This is the effective Date @fcb.param MI, String, pricingTicketNumber, , This is the array facts which are used to calculate the fees @fcb.param MI, String, facilityId, , This is the array facts which are used to calculate the fees @fcb.param MI, String, eventCode, , This is the array facts which are used to calculate the fees @fcb.param MI, Map<String, Object>, fundingTableFacts, , This is the array facts which are used to calculate the fees @fcb.param MI, FundingTableDetailDTO, fundingTableDetailDTO, , This is the array facts which are used to calculate the fees @fcb.param MI, List<String>, pricngPurposeList, , This is the array facts which are used to calculate the fees @return FundingTableDetailDTO, Indicates the Funding table details with all pricing information and other information @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service will fetch origination fees for lending product group being applied by the applicant.  The following activities will be carried out in sequence.  1. Pricing Plan will be created using combination of submission and facility Id as the arrangement number.  2. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  3. This ticket number is stored in Origination to refresh fees related to the event.   4. Pricing module is called to get pricing information for this New Facility or Amend   Facility Event.  @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant information. @fcb.param MI, CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details. @fcb.param MI, CollateralDTO[], collateralDTOs, , Indicates the collateral details. @return {@link PricingResponse} This class holds ticket number for origination fees which is stored.  All inquiries for event associated with facility id is made using this ticket number.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service is used for creating pricing plan for bundle subscription fees.  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for this pricing plan.  This reference Id is generated using the account module and is generated only once   for the given submission. For all product groups under the submission, the same bundle  reference Id is to be used.  2. The Pricing Plan is created using this bundle reference Id.   3. A new pricing ticket number is generated and stored in Origination for Bundle Fees.  4. Pricing Module is called to get pricing information for this Bundle Subscription Event.  Fees can be collected, waived or deferred for this bundle reference Id.   The pricing ticket number stored for bundle fees is used for tracking such   fees collected against this arrangement number. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.  All inquiries for this bundle reference Id are made using this ticket number.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts	This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return @throws FatalException	<p>this pricing plan.&lt;br&gt; This reference Id is generated using the account module and is generated only once &lt;br&gt; for the given submission.&lt;br&gt; 2. The Pricing Plan is created using this bundle reference Id. &lt;br&gt; 3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.&lt;br&gt; 4. Pricing Module is called to get pricing information for this Bundle Subscription Event.&lt;br&gt; 5. If bundle subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.&lt;br&gt; 6. Else Fees can be negotiated, waived or deferred for this bundle reference Id. &lt;br&gt; The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO, ,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.&lt;br&gt; All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service will fetch the fees applied for withdrawing the submission.  The following activities will be carried out in sequence.  1. Pricing Ticket Number is generated and is used for creating price recommendations.  2. Price Recommendations will be created for the submissionId.  3. Pricing Plan will be created using the above submissionId as the arrangement number.  4. A new pricing ticket number is now generated and associated with the event for which  price is to be fetched.  5. This ticket number is stored in Origination to refresh fees related to the event.   6. Pricing module is called to get pricing information for this OR_SUBMISSION_WITHDRAW Event.  @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, submissionId, , Indicates the identifier of the submission @return {@link PricingResponse} This class holds ticket number for withdrawal fees which is stored. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	getPriceOnEvent	This service is used for fetching fees for the event passed.   @param sessionContext @param effectiveDate @param bundleId @param arrangementNumber @param eventCode @param facts @return @throws FatalException	This service will confirm structure solution for lending product group in a submission.   Various operations performed during this service are as follows:   <ol> <li>The facility structure is validated as per the guidelines. {@link LendingApplicationHelper#checkFacilityConformity}</li>  <li>Validations for confirming structure solution are validated. {@link LendingApplicationHelper#validateConfirmActionOnStructureSolution}</li>  <li>Check if ticket number for Origination fees is present. If yes, defer all the uncollected fees for that ticket number else create a new pricing plan, fetch and defer the origination fees.</li>  <li>Check if product group is setup under a bundle and ticket number for bundle fees is present. If yes, defer all the fees for that ticket number else create a new pricing plan, fetch and defer the bundle fees.</li>  <li>In case of Amended facility, check if settlement is required or not (irrespective of the settlement flag on product group and update the same) {@link LendingApplicationHelper#checkIfSettlementRequiredForPG}</li>  <li>For Loan product groups, update party id, borrowing entity and facility details for loan simulations</li>  <li>Check for any rates and fee negotiations. If present, send for discretionary pricing approval</li>  </ol> @param SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param String applicationId - the application for which account has to be opened @param String productGroupSerialNumber - the product group serial number for which account has to be opened @return LendingAccountCreationResponse, this class returns

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the LendingAccountCreationResponse object,  @throws FatalException @exception &lt;br&gt; {@link  OriginationErrorConstants#KYC_DETAILS_NOT_FO  UND} : FC_OR_PI_0001 &lt;br&gt; This check is part of  Pre-create account validation. &lt;br&gt; This exception is  thrown if KYC has not been confirmed for all parties  &lt;br&gt; applying to the product group for which account  is being opened. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#OR_DPA_APPROVAL} :  FC_OR_DE_0082 &lt;br&gt; This exception is raised as an  overridable exception when pricing rates and fees are  negotiated and require approval. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_AMOUNT_  GREATER} : FC_OR_SS_0003 &lt;br&gt; This exception  occurs when one or more sub facilities are configured  and the requested amount for these sub-lines exceed  their parent facility's (can be main facility or another  sub facility) amount. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#REQUESTED_TERM_GRE  ATER} : FC_OR_SS_0004 &lt;br&gt; This exception occurs  when one or more sub facilities are configured and the  requested term for these sub-lines exceed their parent  facility's (can be main facility or another sub facility)  term. &lt;br&gt; &lt;br&gt; {@link  OriginationErrorConstants#CONFIRM_STRUCTURE_  SOLN_FAILED} : FC_OR_SS_0005 &lt;br&gt; This  exception occurs when confirming the structure  solution fails for any reason such as facility structure  is not as per rules or any validations fail &lt;br&gt; &lt;br&gt;  {@link  OriginationErrorConstants#LVR_THRESHOLD_BREA  CHED} : FC_OR_SS_0006 &lt;br&gt; This exception occurs  when the loan to value ratio is breached beyond the  limit allowed for a product group. This threshold  value is maintained in PMU in credit policy template</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>for the product group &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the a single facility or sub-facility is greater than requested amount of that facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUM_OF_ALL_ACCTS_GREATER_THAN_MAIN_FACILITY} : FC_OR_SS_0007 &lt;br&gt; This exception occurs when the sum of requested amounts of all accounts under the facility structure exceed the total amount of main facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#ACCT_NOT_CONFIGURED} : FC_OR_SS_0015 &lt;br&gt; This exception occurs when structure solution is being confirmed with no account configured under the facility. &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#FETCH_PRICE_OFFERINGS_FAILED} : FC_OR_SS_0016 &lt;br&gt; This exception occurs when fetching of price recommendations for origination fees from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PPC_CREATION_FAILED} : FC_OR_SS_0017 &lt;br&gt; This exception occurs when creation of pricing plan for origination fees in pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#GET_PRICE_ON_EVENT_FAILED} : FC_OR_SS_0018 &lt;br&gt; This exception occurs when fetching price for events in origination (OR_NEW_FACILITY or OR_AMEND_FACILITY) from pricing module fails.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OD_FIXED_TERM_GREATER} : FC_OR_SS_0019 &lt;br&gt; This exception occurs when fixed term entered for overdraft product groups</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>exceed the parent requested limit term.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICANTS_HOLDER_PREFERENCES_INCORRECT} : FC_OR_SS_0021&lt;br&gt; This exception occurs when user configures the account configuration in structure solution and then goes back to the product details page and changes the applicants applying for the product group.&lt;br&gt; The user is prompted to check the account configuration again to change the account holder preference for changed applicants.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_FEE_NEGOTIATION} : FC_PR_FEE_OVR &lt;br&gt; This exception occurs any of the rates or fees are negotiated.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#COLLATERAL_VALUATION_INCOMPLETE} : FC_OR_CL_0001 &lt;br&gt; This exception occurs if NCCP is yes and valuation for any collaterals linked to the facility is not yet complete.&lt;br&gt;</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method will set the account Number as the servicing arrangement number in pricing.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is exact copy of intimateServicingArrangement. Only difference is in populateTransactionDetailsDTO, the external reference number is not set explicitly so that the original one is retained (the one that is stamped on confirmSS) OBP23TEST-5044 @param sessionContext @param eventCode @param ticketNumber @param nominatedAccountId @param capitalizedAccountId @param currency @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method will call intimatePricing to update chargeActionType as Capitalized for the Pricing plan for bundle subscription fees @param sessionContext @param facts @param pricingTicketNumber @param pricingFundingItemDTOs @param arrangementNumber @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is called on save of collect fees. The following activities will be carried out in sequence:   1. Call the pricing service to persist the deferred and waived fees.  2. Call the settlement module to collect fees and raise accounting event for the fees to be collected. Component Preference DTO can be null in case there is no fees to be collected.   @param sessionContext @param priceFeatureDTO @param componentPreferenceDTO @param lendingProductGroupLinkageDTO @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is called on save of bundle fees to mark the fees as deferred or waived.  The following activities will be carried out in sequence:  1. Call the pricing service to persist the deferred and waived fees.   2. Raise alert if the account nominated for bundle fees is created in current submission.  @param sessionContext @param priceFeatureDTO @param bundleDetailsDTO @param productGroupLinkageDTO @return @throws



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is called on withdraw submission. The following activities will be carried out in sequence: 1. Call the pricing service to persist the deferred fees if any. 2. Call the settlement module to collect fees and raise accounting event for the fees to be collected. Component Preference DTO can be null in case there is no fees to be collected. @param sessionContext @param priceFeatureDTO @param componentPreferenceDTO @param submissionId @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method will set the given charge action type for all fees against the given ticket no. and intimate pricing. The allowed charge action types are DEFER/WAIVE/CAPITALIZED. It cannot be COLLECT. @param sessionContext @param pricingTicket @param arrangementNo @param currency @param domainCategory @param chargeActionType @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method is used to process Origination and capitalised fees 1. In case of Amend Loan with an account(s) with increased term,offerswaped, nominated and capitalised for fees with the existing account. 2. In case of New Loan with accounts nominated and capitalised for fees after the account creation. OBP23TEST-8963.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimatePricing	This method is called during collect Fees to persist in Pricing the Deferred and Waived Fees. @param priceFeatureDTO @return @throws FatalException	This method will call intimatePricing to update chargeActionType as Capitalized for the Pricing plan for all pricing fees which have been marked as capitalized in funding table @param sessionContext @param facts @param pricingTicketNumber @param pricingFundingItemDTOs @param arrangementNumber @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchDeferredFeesForArrangementNumber	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception	This method checks if there are any uncollected origination fees for the product group. A call is made to pricing with the arrangement number = facilityId @throws FatalException @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId , Indicates the id of the lending application for which the create offer stage is in progress @return List<ValidationError>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchDeferredFeesForArrangementNumber	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception	@param prodGrpLinkage @param nominatedAccountId @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchDeferredFeesForArrangementNumber	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchDeferredFeesForArrangementNumber	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception	This method retrieves the deferred fees which are computed but not charged for a given arrangement number. @param currencyCodes @param arrangementNumber @param sessionContext @return an array of type {@link PriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			exception	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This method is used to fetch origination fees using the application id and facility id. The logic of the method is as follows   <ol> <li>Fetch pricing ticket number using the service @link{OriginationFeeApplicationService#fetchOriginationFees(SessionContext, HeaderDTO, LendingProductGroupLinkageDTO, ApplicantDataDTO, CreditFacilityDTO[], CollateralDTO[])}.  The input of This service can be derived as follows,  <ul> <li>SessionContext - SessionContext from the input HeaderDTO - Inquire application details using base application service @link{com.ofss.fc.app.origination.service.core.application.BaseApplicationService.inquireApplication} using the application id, input. Pass the header dto from the ApplicationDTO retrieved.</li> <li>LendingProductGroupLinkageDTO - Fetch product group linkage from the ApplicationDTO retrieved above, using the facility id, input. Set the PricingTicketNumberForBundleFees to the pg linkage dto, if IsNominatedForBundle() and PricingTicketNumberForBundleFees() is empty, by getting the new pricing ticket number by invoking the service @link{OriginationFeeApplicationService#createBundleSubscriptionPricingPlan }</li> <li>ApplicantDataDTO <li>Fetch all applicants for the application using the helper method @link{LendingApplicationHelper#fetchAllApplicantsForApplicationId }</li> <li>Convert applicant DTO To applicant data DTO using the helper method @link{SubmissionInquiryHelper#populateApplicantD

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ataDTO &lt;/li&gt;&lt;/li&gt; &lt;li&gt;CreditFacilityDTO[] - Fetch credit facility array DTO using the service</p> <p>@link{LendingSimulationApplicationService#fetchFacilityHierarchy &lt;/li&gt; &lt;li&gt;CollateralDTO[] - Fetch collateral using the service @link{LendingApplicationService#fetchCollateralsForApplication}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;</p> <p>&lt;li&gt;Use the pricing ticket number and get the pricing details using the service</p> <p>@link{OriginationPricingAdapter#fetchFeesForTicketNumber . The input facts can be populated using the methods</p> <p>@link{LendingApplicationService#populateFactsForOriginationFee) and</p> <p>@link{LendingApplicationService#populateFundingFactDefaultValues &lt;/li&gt; &lt;li&gt;Iterate</p> <p>WrappedPriceFeatureInquiryResponse</p> <p>#wrappedPriceFeatureDTO[], convert the</p> <p>WrappedPriceFeatureDTO to PricingFeeDetailDTO using the builder</p> <p>@link{PricingFeeDetailDTOBuilder#build &lt;/li&gt;</p> <p>&lt;li&gt;Set the pricing ticket number and PricingFeeDetail DTOs in the response&lt;/li&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc</p> <p>@fcb.param MI,String,applicationId, This object holds the application id, for which the origination fees is to be retrieved @fcb.param MI,String,facilityId, This object holds the facility id, for which the origination fees is to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.pricing.PricingFeesResponse}, This contains the information on origination fees (pricingTicketNumber and pricingFeeDetails) @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.infra.exception.FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	Added this method to fix OBP23TEST-7776 The bundle subscription fee was not getting processed so far. @param sessionContext @param dto @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	@param sessionContext @param dto @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	@param sessionContext @param lendingPGDTOList @param pricingFeesPGMap @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This service will fetch the fees associated with the ticket number and update it back to the funding table @param sessionContext @param fundingTableDetailsDTO @param currency @param pricingTicketNumber @param pricingPurposeList @param facts @return FundingTableDetailDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This method will call intimatePricing to update chargeActionType as Capitalized for the Pricing plan for bundle subscription fees @param sessionContext @param facts @param pricingTicketNumber @param pricingFundingItemDTOs @param arrangementNumber @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This method will set the given charge action type for all fees against the given ticket no. and intimate pricing. The allowed charge action types are DEFER/WAIVE/CAPITALIZED. It cannot be COLLECT. @param sessionContext @param pricingTicket @param arrangementNo @param currency @param domainCategory @param chargeActionType @return @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	Fetch Pricing Fees @param sessionContext @param facts @param pricingTicketNumber @return WrappedPriceFeatureInquiryResponse
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This will update the Bundle Funding Item with the bundle fees @param sessionContext @param fundingItemWithPricing @param lendingPgLinkageDTO @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeesForTicketNumber	This service will fetch the fees associated with the ticket number @param sessionContext @param facts @param pricingTicketNumber @return @throws FatalException	This method will call intimatePricing to update chargeActionType as Capitalized for the Pricing plan for all pricing fees which have been marked as capitalized in funding table @param sessionContext @param facts @param pricingTicketNumber @param pricingFundingItemDTOs @param arrangementNumber @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchBasePricesForTicketNumber	This service will fetch the fees associated with the ticket number and update it back to the funding table @param sessionContext @param fundingTableDetailsDTO @param currency @param pricingTicketNumber @param facts @return FundingTableDetailDTO @throws FatalException	update pricingfee @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchBasePricesForTicketNumber	This service will fetch the fees associated with the ticket number and update it back to the funding table @param sessionContext @param fundingTableDetailsDTO @param currency @param pricingTicketNumber @param facts @return FundingTableDetailDTO @throws FatalException	update pricingfee @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimateServicingArrangement		@param sessionContext @param application @param prodGrpLinkage @param pgSet @param accountOpeningDetailsDTO @param nominatedOffer @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimateServicingArrangement		This method is exact copy of updateNominatedAccountLinkageAndIntimatePricing . Only difference is in populateTransactionDetailsDTO, the external reference number is not set explicitly so that the original one is retained (the one that is stamped on confirmSS) @param sessionContext @param application @param prodGrpLinkage @param pgSet @param accountOpeningDetailsDTO @param nominatedOffer @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	intimateServicingArrangementForSettlement		This method is exact copy of updateNominatedAccountLinkageAndIntimatePricing . Only difference is in populateTransactionDetailsDTO, the external reference number is not set explicitly so that the original one is retained (the one that is stamped on confirmSS) @param sessionContext @param application @param prodGrpLinkage @param pgSet @param accountOpeningDetailsDTO @param nominatedOffer @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	saveNegotiations		This service will save the fee negotiations and capitalization details in Pricing module for all lending PG's .This method is will be called on click of Continue button from Funds Table page for the last lending product group in the submission.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,LendingProductGroupLinkageDTO,lendingProductGroupLinkageDTO, This parameter contains all the information related to lending product group linkage.  @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for <i>Transaction Failure</i> section
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	saveNegotiations		This method saves the discounted/negotiated amount for the PD in Pricing module @param sessionContext @param pricingFundingItemDTOs @param lendingProductGroupLinkageDTO @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	saveNegotiations		This method saves the discounted/negotiated amount for the PD in Pricing module @param sessionContext @param bundleFundingItemDTOs @param lendingProductGroupLinkageDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	addUDMto Recommendation		This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	addUDMto Recommendation		This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration</p> <p>@fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application</p> <p>@fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit.</p> <p>@throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	addUDMto Recommendation		This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	addUDMto Recommendation		This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception</p> <p>&lt;br&gt; {@link OriginationErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeeNegotiationApprovalItemDetails		<p>This method will fetch the fees that were waived or negotiated during account configuration. For CASA/TD/Loan account and Facility fees, the details are fetched from recommendation. For Bundle and Origination fees, the details are fetched from plan.</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeeNegotiationApprovalItemDetails		<p>This method will fetch the the fees that are negotiated or waived during account configuration from pricing and raise overridable exception to send for approvals. @param sessionContext @param lendingProductGroupLinkageDTO @param creditFacilities @param submissionId @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	fetchFeeNegotiationApprovalItemDetails		This method will fetch the the fees that are negotiated or waived during account configuration from pricing and raise overridable exception to send for approvals. @param sessionContext @param productGroupLinkage @return
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	resetFeeNegotiation		
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	resetFeeNegotiation		This method saves the discounted/negotiated amount for the PD in Pricing module @param sessionContext @param pricingFundingItemDTOs @param lendingProductGroupLinkageDTO @throws FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	isTicketReversed		@param sessionContext @param lendingPGDTOList @param pricingFeesPGMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	isTicketReversed		<p>This service will fetch origination fees for lending product group being applied by the applicant.&lt;br&gt;The following activities will be carried out in sequence.&lt;br&gt; 1. Pricing Plan will be created using combination of submission and facility Id as the arrangement number.&lt;br&gt; 2. A new pricing ticket number is now generated and associated with the event for which&lt;br&gt; price is to be fetched.&lt;br&gt; 3. This ticket number is stored in Origination to refresh fees related to the event. &lt;br&gt; 4. Pricing module is called to get pricing information for this New Facility or Amend &lt;br&gt; Facility Event.&lt;br&gt; @fcb.param MI, SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO, applicantDataDTO, , Indicates the primary applicant information. @fcb.param MI, CreditFacilityDTO[], creditFacilityDTOs, , Indicates the facility details. @fcb.param MI, CollateralDTO[], collateralDTOs, , Indicates the collateral details. @return {@link PricingResponse} This class holds ticket number for origination fees which is stored.&lt;br&gt; All inquiries for event associated with facility id is made using this ticket number.</p>
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	isTicketReversed		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	isTicketReversed		This service update all funding tables in submissionInputDTO with pricing details from pricing module. The response will contain updated pricing details that might have been modified through pricing. This service should be called pre assessment where the funding table details is not persisted. LMI amount will also be refreshed with the value from CapitalizationDetailsDTO. If there's no capitalization for LMI, this method will not update the LMI amount @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  @fcb.param MI,submissionInputDTO contain submission information @return FundingTableDetailInquiryResponse, Returns the funding table details dto with refreshed pricing details @throws com.ofss.fc.infra.exception.FatalException
IOrigination PricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPricingAdapter	updateUDMStatus		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerMemo		The method checks if the customer has a memo. Internally the method connects with the party adapter and checks if the customer has a memo. Refer {@link IDDAPartyAdapter#fetchCustomerMemo}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerMemo		The method checks if the customer has a memo. Internally the method connects with the party adapter and checks if the customer has a memo. Refer {@link IDDAPartyAdapter#fetchCustomerMemo}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerMemo		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerMemo		The following method is used to fetch account memo related details.The following method is used to avoid interaction open and close related operations. @param accountId @param response @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerMemo		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method will add PayeeName if it is null from portal side. @param sessionContext @param dto @param componentPreferenceDTO @param accountService @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method will fetch the nick name of the primary party linked to the account. @param accountId @param partyId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		The following emthdo is used to fetch the party account relationship; @param partyId @param nickName @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This method fetches all the instructions and does the filtering on the instructions based on the search criterias. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param partyId, String MI, denotes the party Id. @return FundsManagementSetupResponse @throws com.ofss.fc.infra.exception.FatalException Please

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				refer to the documentation above for <i>Transaction Failure</i> section
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This is a private method which performs the actual fetch and conversion of Core DTO into Basic DTO's. @param sessionContext MI, SessionContext, sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param partyId MI, String, this attribute denotes party identifier for which the sweep instructions to be fetched. @return SweepInstructionInquiryResponse SweepInstructionInquiryResponse, contains two arrays. <ol style="list-style-type: none"> <li>SweepInInstructionBasicDTO, Stores basic details of the Sweep In Instructions.</li> <li>SweepOutInstructionBasicDTO, Stores basic details of the Sweep Out Instructions.</li> </ol> @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This method will be used for inquiring the overdraft limit details of all the accounts for a party. The method works as follows: <ul style="list-style-type: none"> <li>Inquire all the accounts for a party by calling the service, {@link DemandDepositAccountService#fetchAllAccountsForAParty}</li> <li>For each of these accounts, call the method {@link DemandDepositApplicationService#fetchOverdraftLimit}</li> <li>Populate the details in the final response, only if a facility is linked to the account.</li> </ul> @param sessionContext-This parameter captures the basic information about the user of the system @param partyId-Identifies Party. @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This method will fetch all Accounts related to Borrowing Id. @param sessionContext @param borrowingEntityCode @return accountArray @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This method fetch all the DemandDepositAccounts (i.e. all the accounts of a party with AccountType = CASA) for a party. This method works as follows :  <pre> List<String> accountList = new ArrayList<String>(); PartyToAccountAllRelationshipsInquiryResponse response = {@link IPartyToAccountRelationshipApplicationService#fetchAllAccountsOfParty()} for(PartyAccountRelationshipDTO partyAccountRelationshipDTO = response.getPartyToAccountRelationshipDTOs()) { if(partyAccountRelationshipDTO.getAccountType == AccountType#CASA) { accountList.add(partyAccountRelationshipDTO.getPartyAccountRelationshipKeyDTO().getAccountId()) } } </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				return accountList </pre> @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	An account closure request simply marks the account for closure and does not actually close the account. The account closure request is subsequently evaluated by a relationship manager who eventually decides whether to go ahead with the actual closure or retain the account after talking to the customer. <p> After the account is marked for closure', user initiated debits are not permitted. System (interest & fees) initiated debits are permitted after the account is marked for closure'. </p> <p> System (interest and rewards) and user (funds transfer, cash deposit etc.) initiated credits are permitted to the account after the account is marked for closure'. </p> <ul> @param sessionContext @param accountId @param primaryReasonCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	This method returns the array of all the instructions maintained for all accounts a party.  For a single account multiple instructions can be maintained.  All the demand deposit accounts of a party are first retrieved from party module. Then the individual instruction lists for the accounts are fetched. This method works as follows :   <pre> accountList = {@link IDDAPartyAdapter#fetchAllDemandDepositAccountsOfParty()} instructionDTOs[]; for(all accountList) { sweepInInstructionDTOs = {@link SweepInInstructionApplicationService#fetchAllSweepInInstructionsForBeneficiary()} add sweepInInstructionDTOs to instructionDTOs. } for(all accountList) { sweepOutInstructionDTOs = {@link SweepOutInstructionApplicationService#fetchAllSweepOutInstructionsForAccount()} add sweepOutInstructionDTOs to instructionDTOs. } for(all accountList) { standingInstructionDTOs = {@link StandingInstructionApplicationService#fetchAllStandingInstructionsForAccount()} add standingInstructionDTOs to instructionDTOs. } add instructionDTOs[] to InstructionInquiryResponse. <pre> &#064;param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. &#064;param partyId &#064;return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllDemandDepositAccountsOfParty	This method returns the list of all demand deposit accounts of a party. @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method will inquire party details for party id. @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method is used to inquire the details of each member of the group. The method works as follows: <ul> <li>Load the account for the input accountId.</li> <li>Load the party details for the input accountId to get the party name.</li> <li>Populate the memeber details DTO.</li> @param accountGroup @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method is used to get the details of the party used for the report generation for the newly opened account It populates the default address(Home address) of the party, the mobile contact details and the email Id @partyId



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method is used to fetch the details of party depending on the party id specified @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		Method will populate details of BasicInquiryDetailsDTO using entity {@link IDemandDepositAccount}; @param demandDepositAccount @return BasicInquiryDetailsDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method is used to populate the value of party address The default home address is set for the party. @param demandDepositAccount @param demandDepositAccountReport
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetails		This method fetches the party name from given party id @param partyId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	This method is responsible for calling the party module inorder to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	This method is responsible for calling the party module inorder to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	<p>&lt;ol&gt; All accounts can have two type of statuses &lt;li&gt; {@link SystemDefinedStatus} &lt;/li&gt; &lt;li&gt; {@link UserDefinedStatus} &lt;/li&gt; &lt;/ol&gt; The SystemDefinedStatus at the time of account opening will be defaulted to Regular. But the user defined status may vary based on kyc status and documentation completed. The user defined status will be based on the following matrix. &lt;table class="tableizer-table" border="1"&gt; &lt;tr class="tableizer-firstrow"&gt; &lt;th&gt;User Defined Account Status&lt;/th&gt; &lt;th&gt;KYC Confirmed&lt;/th&gt; &lt;th&gt;Completed documentation&lt;/th&gt; &lt;th&gt;Reason&lt;/th&gt; &lt;th&gt;Reason Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;NULL&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Debit Block&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;KYC Incomplete&lt;/td&gt; &lt;td&gt;RC1&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Debit Block&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;Document Incomplete&lt;/td&gt; &lt;td&gt;RC2&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Debit Block&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;KYC Incomplete&lt;/td&gt; &lt;td&gt;RC1, RC2&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;li&gt; Call {@link this.isKYCStatusConfirmed()} for getting the kyc status. If the method returns false then KYC for the party is UNCONFIRMED. UserDefinedStatus will be as metioned in the above table. &lt;/li&gt; &lt;li&gt; The documentation completion flag is there in the input. Origination will provide the details regarding the completeness of documents required for account opening (Origination will give an indicator for the same). If the documentation is not complete, system will open the account with the status as ␣ Debit Block␣ with reason as ␣ Document Incomplete␣.&lt;/li&gt; &lt;li&gt; Reason code is defined in the ba_reason_codes&lt;/li&gt; Create instances of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				UserDefinedStatus for each reason i.e. kyc unconfirmed and incomplete documentation. The fields of UserDefinedStatus are populated as follows: reasonCodes = RC1 effectiveDate = postingDate processDate = postingDate DemandDepositLifecycleStatus = DebitBlocked Add each instance to the DemandDepositAccount.Set<UserDefinedStatus> @param demandDepositAccount @param postingDate @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	<p>&lt;ol&gt; If the party status of any of the holders is <math>\cap \uparrow \Downarrow</math> KYC Unconfirmed <math>\cap \uparrow \Downarrow</math>, system will open the CASA account with status as <math>\cap \uparrow \Downarrow</math> Debit Block <math>\cap \uparrow \Downarrow</math> with reason as <math>\cap \uparrow \Downarrow</math> KYC Incomplete <math>\cap \uparrow \Downarrow</math>. The algorithm is as follows: &lt;li&gt;Get instance of the DDAPartyAdapter using AdapterFactory&lt;/li&gt; &lt;li&gt;Call {@link DDAPartyAdapter#isKYCConfirmed}&lt;/li&gt; &lt;li&gt;return the kycConfirmedFlag boolean back&lt;/li&gt; &lt;/ol&gt; @param partyId @param kycConfirmationFlag @throws FatalException</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isKYCConfirmed	This method is used to check the KYC @param partyId	This method will be used to populate the DTO for activity log in case of modify account status is changed. @param accountId @param object @param sessionContext @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isPartyExistWithRole	This method is whether the given party exists with the specified role @param partyId @param roleid	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	This method validates the input, fetches the account and the minor account status corresponding to the new primary holder and returns the updated account. @param modificationDTO @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	This method is used to fetch the party status whether it is Major, Minor or Operational_Major The different status combination possible are 1. Minor in case if both operational major and legal major is false 2. Operational major : In case if operational major is true and legal major is false 3. Legal major : In case if legal major is true and operational major is false If both legal and operation major is true it is an invalid case. @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyMinorStatus	This method is used to fetch the party minor status @param partyId	The minor status for an account is derived based on conditions mentioned below. <table class="tableizer-table" border="1"> <tr class="tableizer-firstrow"> <th>Sr No</th> <th>Primary Party Minor-Major Status</th> <th>Secondary Party Available</th> <th>Secondary Party Minor-Major Status</th> <th>Account Opening Permitted</th> <th>CASA Account Major-Minor Status</th> </tr> <tr> <td>1</td> <td>Minor</td> <td>No</td> <td>NA</td> <td>&#8730;</td> <td>Minor with restrictions</td> </tr> <tr> <td>2</td> <td>Minor</td> <td>Yes</td> <td>Minor</td> <td>&#8730;</td> <td>Minor with restrictions</td> </tr> <tr> <td>3</td> <td>Minor</td> <td>Yes</td> <td>Operational Major</td> <td>&#8730;</td> <td>Minor with restrictions</td> </tr> <tr> <td>4</td> <td>Minor</td> <td>Yes</td> <td>Legal Major</td> <td>&#8730;</td> <td>Minor with restrictions</td> </tr> <tr> <td>5</td> <td>Operational Major</td> <td>No</td> <td>NA</td> <td>&#8730;</td> <td>Minor without restrictions</td> </tr> <tr> <td>6</td> <td>Operational Major</td> <td>Yes</td> <td>Operational Major</td> <td>&#8730;</td> <td>Minor without restrictions</td> </tr> <tr> <td>7</td> <td>Operational Major</td> <td>Yes</td> <td>Legal Major</td> <td>&#8730;</td> <td>Minor without restrictions</td> </tr> <tr> <td>8</td> <td>Legal Major</td> <td>No</td> <td>NA</td> <td>&#8730;</td> <td>Major</td> </tr> <tr> <td>9</td> <td>Legal Major</td> <td>Yes</td> <td>Minor</td> <td>&#8730;</td> <td>Major</td> </tr> <tr> <td>10</td>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt;  &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;&amp;#8730;&lt;/td&gt;  &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;11&lt;/td&gt;  &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal  Major&lt;/td&gt; &lt;td&gt;&amp;#8730;&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt;  &lt;/tr&gt; &lt;/table&gt; Minor-major status of the account on  account opening will always depend on the Primary  holder in the account. Call  IDDAPartyAdapter#fetchParty(partyId): AgeStatus  (party minor status) @param  partyToAccountRelationship @throws FatalException</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerBlacklisted		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchCustomerBlacklisted		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not @param partyId	The method checks if the customer is blacklisted. Internally, the method calls party adapter and checks if the party is blacklisted. Refer {@link ILoanPartyInformationApplicationAdapter#isCustomerBlacklisted}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not @param partyId	The method checks if the customer is blacklisted. Internally the method connects with the party adapter and checks if the customer is blacklisted. Refer {@link IDDAPartyAdapter#isCustomerBlacklisted}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not @param partyId	The following method is used to validate account related information. The following method has been added to avoid interaction begin and close related calls. @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	createPartyAccountRelation	This method is used to store the details of all the parties @param partyAccountRelationshipAdapterDTO @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.createNewPartyAccountRelationship(SessionContext, PartyAccountRelationshipDTO) which is used to store the details of all the parties at the time of account Opening. @param partyAccRelationshipAdapterDto
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	createPartyAccountRelation	This method is used to store the details of all the parties @param partyAccountRelationshipAdapterDTO @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	createPartyAccountRelation	This method is used to store the details of all the parties @param partyAccountRelationshipAdapterDTO @throws FatalException	This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method validates all the party id's are linked to the account.   <b>Since:</b> R2.2   @param accountAddressLinkageDTO @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This will create a borrowing entity id.And stamp it on account. @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationship that exists @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationship that exists @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	Returns all the parties related to the accountId Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility. An existing demand deposit account can be used only if the following conditions are fulfilled:    Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners.    <b>For e.g.</b> : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list.     Also, existing CASA selected, must not be linked to any other Overdraft facility.    Refactored to this module from {@link LendingApplicationService} @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @fcb.param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception   {@link DemandDepositAccountErrorConstant#DDA_OFFER_INVALID} : FC_OR_DE_0081   This error occurs when:  The offer details are being fetched for incorrect offer identifier or offer has expired.    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	Returns all the parties related to the accountId Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc														
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	<p>TODO: Add order by partyId in the HQL fetchAllPartiesOfAccount. This method will be used to validate the accounts of the group. The accounts should satisfy the following conditions to be a part of the group.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Should be a valid CASA Account.&lt;/li&gt;</li> <li>&lt;li&gt;Account Status should be valid for Group Account Maintenance.&lt;/li&gt;</li> <li>&lt;li&gt;All the accounts of the group should have the same currency.&lt;/li&gt;</li> <li>&lt;li&gt;The customer account relationship should be valid for Group Account Maintenance.&lt;/li&gt;</li> <li>&lt;li&gt;The account should not belong to more that one group with the same purpose.&lt;/li&gt;</li> <li>&lt;li&gt;There should be a minimum of 2 accounts involved in a group.&lt;/li&gt;</li> <li>&lt;li&gt;The relationship between the account and the party should be in the role of a Customer and should be one of the following: <ul style="list-style-type: none"> <li>&lt;li&gt;Sole Owner&lt;/li&gt;</li> <li>&lt;li&gt;Joint And First&lt;/li&gt;</li> <li>&lt;li&gt;Joint And Other&lt;/li&gt;</li> </ul> </li> <li>&lt;li&gt;All the accounts of the group should be belong to the same party or set of parties as the case maybe.&lt;/li&gt;</li> </ul> <p>For e.g.</p> <table border="1"> <tr> <td>AccountNumber</td> <td>PartiesToTheAccount</td> </tr> <tr> <td>111</td> <td>A~B</td> </tr> <tr> <td>222</td> <td>A~B</td> </tr> <tr> <td>333</td> <td>A~C</td> </tr> <tr> <td>444</td> <td>A~B</td> </tr> <tr> <td>555</td> <td>A~C</td> </tr> <tr> <td>666</td> <td>A</td> </tr> </table> <p>In the above case:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Accounts 111, 222 and 444 can belong to the same group.&lt;/li&gt;</li> <li>&lt;li&gt;Accounts 333 and 555 can belong to the same group.&lt;/li&gt;</li> <li>&lt;li&gt;Accounts 111,222, and 555 cannot belong to the same group.&lt;/li&gt;</li> <li>&lt;li&gt;Accounts 111 and 666 cannot belong to the same group.&lt;/li&gt;</li> </ul> <p>The method works as follows:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Declare</li> </ul>	AccountNumber	PartiesToTheAccount	111	A~B	222	A~B	333	A~C	444	A~B	555	A~C	666	A
AccountNumber	PartiesToTheAccount																	
111	A~B																	
222	A~B																	
333	A~C																	
444	A~B																	
555	A~C																	
666	A																	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>a local variable previousListOfParties = null.&lt;/li&gt;  &lt;li&gt;Declare a local variable previousCurrency = null.&lt;/li&gt; &lt;li&gt;Check the size of the array AccountGroupMember in the business policy DTO. If the size &lt; 2, then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Open a loop for the accounts of maintained for the group. &lt;li&gt;Load the account by calling repository method {@link DemandDepositAccountRepository#load} to validate if the account exists or not. If the account does not exist, then add an error message to the list of validation errors. &lt;/li&gt; &lt;li&gt;If the (previousCurrency &lt;&gt; null &amp;&amp; DemandDepositAccount.getCurrencyCode() &lt;&gt; previousCurrency), then add error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Set the value of previousCurrency = DemandDepositAccount.getCurrencyCode()&lt;/li&gt; &lt;li&gt;Check the status of the account by calling the service {@link DemandDepositAccount#validateStatus} to check if the status is valid for Group Account Maintenance. If the status is not valid, add error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Get the parties associated with the account along with their relationships by calling the service {@link IDDAPartyAdapter#fetchAllPartiesOfAccount}.&lt;/li&gt; &lt;li&gt;Open a loop for the response returned by the above service. &lt;ol&gt; &lt;li&gt;If the looprecord.partyId = accountGroup.partyId, then &lt;ol&gt; &lt;li&gt;If the looprecord.relationshipType is not one of the relationships mentioned above, then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;If the looprecord.roleId &lt;&gt; Customer, then add appropriate error message to the list of validation</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>errors.&lt;/li&gt; &lt;/ol&gt; &lt;li&gt;Concatenate the partyId in a tilde separated list and assign to variable listOfParties.&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;If none of the parties in the list = accountGroup.partyId, then add appropriate error message to the list of Validation errors.&lt;/li&gt; &lt;li&gt;If (previousListOfParties != Null &amp;&amp; previousListOfParties &lt;&gt; listOfParties), then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Set previousListOfParties = listOfParties.&lt;/li&gt; &lt;li&gt;Call the repository method {&amp;@link AccountGroupRepository#fetchNumberOfGroupsFor APurpose} to get the number of groups of the same purpose of which the account is a member. If the returned value &gt; 0, the add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;/li&gt; &lt;/ul&gt;</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	This method is used to fetch all the parties for the given account id @param accountId @return @throws FatalException	This method will be used to populate the DTO for activity log in case of modify account status is changed. @param accountId @param object @param sessionContext @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isPartyDeceased	@param partyId @return @throws FCRException	If the party status of any of the holders is <code>Blacklisted</code> , system will not permit CASA account opening. Call <code>IDDAPartyAdapter#isCustomerBlacklisted(partyId: String)</code> to validate if the party is to be blacklisted. Reject the transaction if the party is blacklisted i.e. add this as a validation error.
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isPartyDeceased	@param partyId @return @throws FCRException	This method is used to check the existence of customer for a specific partyId.
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isPartyDeceased	@param partyId @return @throws FCRException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	This service is used to update the Account Holder preferences for an already existing demand deposit account.  In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,DDAccountHolderPreferenceDTO,preferenceDTO, , This class contains information about preferences captured for the applicant of the demand deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, demand deposit statement delivery preferences and debit card preferences such as color , type, etc. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relationship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartiesOfDdAccount	This service will fetch all the existing parties that are holders for the demand deposit account. @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	This method will be used to update channel facilities in party account relationship details maintenance. Whenever there is a change in channel facilities through account preferences screen, same need to be updated in account party relationship. @param sessionContext @param existingDemandDepositAccount @param bankingFacilitiesDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationship	This service will change the party to account relationship type for each party provided as input @param sessionContext @param accountRelationshipDTOArray @throws FatalException	While updating P2A relationship.Business policy checks for operating instructions are from linked parties only. In case of borrower change in validate mode.If we do not close operating instructions of non borrower call was failing.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationship	This service will change the party to account relationship type for each party provided as input @param sessionContext @param accountRelationshipDTOArray @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.createNewPartyAccountRelationship(SessionContext, PartyAccountRelationshipDTO) which is used to store the details of all the parties at the time of account Opening. @param partyAccRelationshipAdapterDto
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationship	This service will change the party to account relationship type for each party provided as input @param sessionContext @param accountRelationshipDTOArray @throws FatalException	This service is used to update the Account Holder preferences for an already existing demand deposit account.  In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,DDAccountHolderPreferenceDTO,preferenceDTO, , This class contains information about preferences captured for the applicant of the demand deposit account. It has information about the external banking

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				facilities such as internet banking, phone banking and mobile banking, demand deposit statement delivery preferences and debit card preferences such as color , type, etc. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relationship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationship	This service will change the party to account relationship type for each party provided as input @param sessionContext @param accountRelationshipDTOArray @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getPartyDetails	This method is used to get the partydetails based on the partyId @param partyId @return	@param trusteeType @param trusteePartyId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getPartyDetails	This method is used to get the partydetails based on the partyId @param partyId @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getPartyDetails	This method is used to get the partydetails based on the partyId @param partyId @return	This method is used to save the details of the newly opened accounts ,which is used for the report generation. @param demandDepositAccount @param pricingPlanId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isAccountRelatedToParty	This method is to determine whether party and account are related to each other. {@link PartyAccountRelationshipType} are {@link PartyAccountRelationshipType#Joint_And_First}, {@link PartyAccountRelationshipType#Sole_Owner} and {@link PartyAccountRelationshipType#Joint_And_Other} @param accountId @param partyId @return	This method check whether transfer between these accounts is allowed or not. If toAccount is null then it is treated as external account. @param fromAccount @param fromAccountType @param toAccount @param toAccountType @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	isAccountRelatedToParty	This method is to determine whether party and account are related to each other. {@link PartyAccountRelationshipType} are {@link PartyAccountRelationshipType#Joint_And_First}, {@link PartyAccountRelationshipType#Sole_Owner} and {@link PartyAccountRelationshipType#Joint_And_Other} @param accountId @param partyId @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	assignRoleForParty	This method will check if the party has been assigned the role passed. If role is not found, it will assign the new role for the party @param sessionContext @param partyId, The identifier for the party @param roleId, The role which is to be checked for the party @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyRelationship	This service will fetch all existing party - party business relationship for the related party identifier passed @param sessionContext @param partyId, The identifier for the related party @return PartyRelationshipResponse, Contains all party relationships for the related party identifier passed @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetailsForReport	This method will fetch the party details required for the generation of reports @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetailsForReport	This method will fetch the party details required for the generation of reports @param partyId	This method is used to get the details of the party used for the report generation for the newly opened account It populates the default address(Home address) of the party, the mobile contact details and the email Id @partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetailsForReport	This method will fetch the party details required for the generation of reports @param partyId	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	inquirePartyDetailsForReport	This method will fetch the party details required for the generation of reports @param partyId	This method is used to populate the value of party address The default home address is set for the party. @param demandDepositAccount @param demandDepositAccountReport

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	The method validates if there is memo on the account by account.hasMemo(); If account.hasMemo() return true then call {@link MemoService#fetchAllActiveAccountMemos}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create TransactionContext by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate ApplicationService through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method populates the activity log for the Multi fund deposit transaction. @param accountId @param object @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method is used to set the primary PartyId in case of Sole Holer and sets secondaryId in case of Joint Holder. @param activityId @param accountId @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method is used to populate the DTO values for creation of accountaddressLinkage,modification of address account linkage and the deletion of address linkage @param accountId @param object @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	The method validates if there is memo on the account by account.hasMemo(); If account.hasMemo() return true then call {@link MemoService#fetchAllActiveAccountMemos}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	The method checks if the customer has a memo. Internally the method connects with the party adapter and checks if the customer has a memo. Refer {@link IDDAPartyAdapter#fetchCustomerMemo}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	The method validates if there is memo on the account by account.hasMemo(); If account.hasMemo() return true then call {@link MemoService#fetchAllActiveAccountMemos}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	It prepares AccountTransaction and InstrumentDetail.Its input parameters are initiatorAccountId,transactionAmount,chequeNumber,chequeDate @param initiatorAccountId @param transactionAmount @param chequeNumber @param chequeDate @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	The method validates if there is memo on the account by {@link MemoService#fetchAllActiveAccountMemos}
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	Method prepares borrower entity name. It is prepared from all account holders legal name. @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method is used to set the primary PartyId in case of Sole Holer and sets secondaryId in case of Joint Holder. @param activityId @param accountId @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method is used to populate the DTO for transferFundsToBeneficiaries @param object @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	This method will be used to populate the DTO for activity log in case of modify account status is changed. @param accountId @param object @param sessionContext @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyName	This method will fetch legal name of party from flx_pi_names_b.This will call PartyNameApplicationService to get legal name of the party. @param sessionContext @param partyID @return	<p>This method is responsible for populating the details required to raise Alert for addition of hardship relief </p> @param activityId @param accountId @param applicationContext @param object @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyNameForAccount		This method is used to set the primary PartyId in case of Sole Holer and sets secondaryId in case of Joint Holder. @param activityId @param accountId @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyNameForAccount		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyNameForAccount		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyNameForAccount		Method prepares borrower entity name. It is prepared from all account holders legal name. @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyNameForAccount		This method is used to set the primary PartyId in case of Sole Holer and sets secondaryId in case of Joint Holder. @param activityId @param accountId @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	derivePartyNamesOfRelatedParties	Method to derive the party names, if the party is the sole owner of the account, this method will return the first name of that party. If the account is jointly held by multiple parties, the method will return the first name of both the parties. But, if the account is a trust account, then this method will return the name of the trust. @param accountId @return	This method will add PayeeName if it is null from portal side. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param StandingInstructionDTO MI,@link StandingInstructionDTO,StandingInstructionDTO, </br> It contains the following attributes: <ol> <li>key:@link InstructionKeyDTO}MI, It should contain only the accountId.InstructionNumber must be null.It will be generated</li> <li>StandingInstructionAmount:@link Money} This attribute denotes the amount for which the Standing instruction is registered. This amount can be in same currency or different currency as of the account, if its of different currency then the currency conversion rates as of the instruction processing are applicableThis attribute denotes the amount for which the Standing instruction is registered. This amount can be in same currency or different currency as of the account, if its of different currency then the currency conversion rates as of the instruction processing are applicable</li> <li>MinimumBalanceToRetain:@link Money} MI contains the minimum balance to be maintained in the provider account after which the sweep outs are valid.</li> <li>InstructionName: String An instruction is a request to perform some action at some point of time. The instruction can be recurring or one-time in nature.</li>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;nextExecutionDate: {@link RecurrencePattern#calculateNextDate(EpochType, Date)} This attribute denotes the next instruction execution date. This date is always updated when one execution of an instruction is completed. For a newly created instruction the nextDate is always the same as the startDate of the instruction execution. .&lt;/li&gt;</p> <p>&lt;li&gt;MaxInstructions{@link RecurrencePattern#calculateMaxInstructions()} MI :This attribute denotes the number of times the instruction needs to be executed. Either of the two i.e. noOfInstructions or endDate can be provided by the customer while registering an instruction. The other attribute is accordingly derived by the system. If its zero, it mean the instruction will be executed till the account is their or it is not modified.&lt;/li&gt;</p> <p>&lt;li&gt;narrative: String MI This attribute denotes the small description about the instruction that is setup.&lt;/li&gt;</p> <p>&lt;li&gt;MaxRetries :This attribute denotes the maximum number of retries that the system will perform in order to achieve a successful execution of the instruction. The instruction execution will be retried on each consecutive day till the maxRetries is reached.&lt;/li&gt;</p> <p>&lt;li&gt;priority :This attribute denotes the priority of sweep out instruction execution.&lt;/li&gt;</p> <p>&lt;li&gt;From a single provider account the funds might be swept out to more than one beneficiary account, so the order of the sweep outs to those beneficiary accounts will be defined here in this attribute.&lt;/li&gt;</p> <p>&lt;li&gt;For a provider account this number should be unique, as the two sweep out instructions from a provider account need to be unique.&lt;/li&gt;</p> <p>&lt;li&gt;ComponentPreferenceDTO : {@link ComponentPreferenceDTO} the DTO for the ComponentPreference entity.&lt;/li&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;FeeDetailsDTO: {@link FeeDetailsDTO} A transaction/maintenance to be done by the CASA module will need the fee to be calculated by the pricing module.&lt;/li&gt; &lt;li&gt;recurrencePatternDto MI: {@link RecurrencePatternDTO}: It denotes the recurrence pattern maintained for the instruction manually.it cannot be null&lt;/li&gt; &lt;/ol&gt; @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	derivePartyNamesOfRelatedParties	Method to derive the party names, if the party is the sole owner of the account, this method will return the first name of that party. If the account is jointly held by multiple parties, the method will return the first name of both the parties. But, if the account is a trust account, then this method will return the name of the trust. @param accountId @return	This method will fetch the nick name of the primary party linked to the account. @param accountId @param partyId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	derivePartyNamesOfRelatedParties	Method to derive the party names, if the party is the sole owner of the account, this method will return the first name of that party. If the account is jointly held by multiple parties, the method will return the first name of both the parties. But, if the account is a trust account, then this method will return the name of the trust. @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	derivePartyNamesOfRelatedParties	Method to derive the party names, if the party is the sole owner of the account, this method will return the first name of that party. If the account is jointly held by multiple parties, the method will return the first name of both the parties. But, if the account is a trust account, then this method will return the name of the trust. @param accountId @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	derivePartyNamesOfRelatedParties	Method to derive the party names, if the party is the sole owner of the account, this method will return the first name of that party. If the account is jointly held by multiple parties, the method will return the first name of both the parties. But, if the account is a trust account, then this method will return the name of the trust. @param accountId @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	This method check if there are any joint parties for given account with warning indicators @param warningIndicatorTypeForRemoval @param noWarningIndicatorOnAccount @param sessionContext @param dDAPartyAdapter @param partyAccountRelationshipDTO @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	<pre>This method will validate the party id for any warning indicators to be present passed by the Origination module during account opening. There can be more than one party id in the PartyToAccountRelationship[] as there can be joint account opening request. For each party check the party Id has warning indicator. &lt;li&gt; Call {@link IDDAPartyAdapter#fetchWarningIndicatorsOnParty(sessionContext, partyId)};&lt;/li&gt; &lt;li&gt;&lt;ol&gt; isPartyExistWithRole can throw two exceptions &lt;li&gt; check if warning indicator is present on the party id&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; </pre>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	<pre>This method will validate the party id for any warning indicators to be present passed by the Origination module during account opening. There can be more than one party id in the PartyToAccountRelationship[] as there can be joint account opening request. For each party check the party Id has warning indicator. &lt;li&gt; Call {@link IDDAPartyAdapter#fetchWarningIndicatorsOnParty(sessionContext, partyId)}&lt;/li&gt; &lt;ol&gt; isPartyExistWithRole can throw two exceptions &lt;li&gt; check if warning indicator is present on the party id&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; </pre>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchWarningIndicatorsOnParty	This method is used to fetch all the warning indicators for the given party by calling the party warning indicator application service @param sessionContext @param partyID @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		This method is used to fetch the party status whether it is Major,Minor or Operational_Major The different status combination possible are 1. Minor in case if both operational major and legal major is false 2. Operational major : In case if operational major is true and legal major is false 3. Legal major : In case if legal major is true and operational major is false If both legal and operation major is true it is an invalid case. @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllAccountsOfParty		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllAccountsOfParty		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllAccountsOfParty		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	modifyChannelFacilitiesInPartyAccountRelationshipDtls	This method will be used to update channel facilities in party account relationship details maintenance.  Whenever there is a change in channel facilities through account preferences screen, same need to be updated in account party relationship. @param sessionContext @param accountRelationshipDTOs	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	modifyChannelFacilitiesInPartyAccountRelationshipDtls	This method will be used to update channel facilities in party account relationship details maintenance.  Whenever there is a change in channel facilities through account preferences screen, same need to be updated in account party relationship. @param sessionContext @param accountRelationshipDTOs	This method will be used to update channel facilities in party account relationship details maintenance.  Whenever there is a change in channel facilities through account preferences screen, same need to be updated in account party relationship. @param sessionContext @param existingDemandDepositAccount @param bankingFacilitiesDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	setPostalAddressDetails	Sets the postal address details.Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception	This method would create the account address linkage while creating a account @param accountId @param partyId @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	setPostalAddressDetails	Sets the postal address details.Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		@param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		@param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		@param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		@param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		@param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to identify the account number given the Account Nick name and Party Id @param sessionContext @param accountNumber @return NamedPartyRelationshipDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to identify the account number given the Account Nick name and Party Id @param sessionContext @param accountNumber @return NamedPartyRelationshipDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to identify the account number given the Account Nick name and Party Id @param sessionContext @param accountNumber @return NamedPartyRelationshipDTO
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method will add PayeeName if it is null from portal side. @param sessionContext @param dto @param componentPreferenceDTO @param accountService @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method will fetch the nick name of the primary party linked to the account. @param accountId @param partyId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		The following emthdo is used to fetch the party account relationship; @param partyId @param nickName @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		This method is used to fetch all the parties of the account, it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAccountRelationship		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountRelationshipForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountRelationshipForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountRelationshipForNickName		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountRelationshipForNickName		This method is used to fetch the account relationship form the nick name and the party id provided. @param nickName @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	This method is used to fetch the account number for the nick name and the party id provided.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method is used to fetch the account number for the nick name and the party id provided.
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method is used to fetch the account relationship for the nick name and the party id provided.
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method is used to fetch the account details for input account number and account type.   <i>Both accountId and accountType are mandatory inputs to this service.</i>  Any one of the following can be supplied in AccountType: <ul><li>Loan</li> <li>Term_Deposit</li><li>CASA</li> <li>Credit_Card</li> </ul>This service returns details of the account which includes title, nick name, status, branch, currency, offer, opening date, holder name, balance details, business unit and other account specific data like clear balance, unclear balance in case of CASA accounts, disbursement, schedule, installment, maturity details in case of LOAN accounts. (For detailed information please refer {@link com.ofss.fc.app.adapter.account.dto.AccountDTO})    Additionally, this service checks if there are any memos attached to this account. If found, it will be returned in the {@link AccountInquiryResponse#getStatus()#getReplyText()} as "The account memo is present on 'account number'. Description: 'memo description'".  If agent is linked to the account, this service also returns the agent details like agentId, agentType, agentName, commissionCategoryType, email, phone (For more, please check {@link com.ofss.fc.app.adapter.account.dto.AgentDetailsDTO}) @param sessionContext @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method should be used by all internal modules for inquiring the account. This will avoid Interaction begin and close related calls. @param sessionContext @param accountId @param accountType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method is used to fetch the account details for input account number.   <i>AccountId is a mandatory input to this service.</i>  It will first inquire the type of account and then invoke {@link #fetchAccountByAccountType(SessionContext, String, AccountType)} to fetch account specific details. @param sessionContext @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	The method will validate to check if the transaction amount requires transacting party details are not. It would also return all the operating instructions that are applicable for the transaction amount specified. <p> Incase the amount is more than the threshold amount, the method will also set the reply code to {@link ResponseCodeType#OVR_EXCEP_PRES_01}. </p> @param sessionContext @param arrangementCode @param transactionAmount @param eventId @param accountType @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAccountByAccount	@param accountId @param partyId @return @throws FatalException	This method is used to close Deposit. @param accountId Account id to which deposit belongs @param depositId Identifier of Deposit to be closed @return {@link TransactionStatus} containing result of close operation @see com.ofss.fc.app.td.service.deposit.IDepositApplicationService#closeDeposit(com.ofss.fc.app.context.SessionContext, java.lang.String, int)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAllDetailsFromMDMParty	This method will fetch all party details with identification details. @param partyId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAllDetailsFromMDMParty	This method will fetch all party details with identification details. @param partyId @return @throws FatalException	This method will fetch transacting party details for given input. @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyForSearch	@param partyId @param searchList @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyForSearch	@param partyId @param searchList @return	This represents the toFetchTransactingPartyForAccount method of the Transacting party application service. This method will fetch party details for party. @param MI,PartyAccountRelationshipAdapterDTO,partyAccountRelationshipAdapterDTO above @return {@link com.ofss.fc.domain.account.entity.transactingparty.TransactingParty} @return {@link TransactingPartyDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		This validation will do following: Check the minor account status of the account The issuance of chequebook to a party depends on the Minor Account Status @link com.ofss.fc.enumeration.dda.MinorAccountStatus}. There are three possible statuses at account level. 1) Major 2) MinorWithoutRestrictions 3) MinorWithRestrictions 1)Major: It will

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>allow to perform the cheque book request.&lt;br /&gt; &lt;br /&gt; 2)MinorWithoutRestrictions: &lt;br /&gt; There are two parameters at the bank policy level.&lt;br /&gt; a) relation required&lt;br /&gt; b) relation type&lt;br /&gt; &lt;br /&gt; When relation Flag = FALSE&lt;br /&gt; If the relation required flag is false then there is no restriction for this status and cheque book request will be honored.&lt;br /&gt; &lt;br /&gt; When relation Flag = TRUE&lt;br /&gt; Validate if the relation type defined at the bank policy level is present in the party account relationship for this account.&lt;br /&gt; &lt;br /&gt; If present then allow cheque book request.&lt;br /&gt; &lt;br /&gt; If account DOES NOT have relationship present then &lt;br /&gt; throw business validation excetion.&lt;br /&gt; &lt;br /&gt; For example relation required flag is TRUE and relationship defined is Guardian then this account should have a guardian relation ship maintained for the account.&lt;br /&gt; &lt;br /&gt; 3)MinorWithRestrictions: &lt;br /&gt; There are two separate paramters maintained at bank policy level for this status as well. The rest of the processing is exactly same as MinorWithoutRestrictions.&lt;br /&gt; &lt;br /&gt; The derivation of the Account Minor status is created at account opening itself and it is updated if a parties minor status changes in batch.&lt;br /&gt; &lt;br /&gt; &lt;br /&gt; Validation for auto reorder of chequebook.&lt;br /&gt; The same rules of chequebook request for minor status will also be applicable for reorder of chequebook.&lt;br /&gt; @param account</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		@param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		<pre> This method opens a DDA by cloning an existing account. The account is opened by calling {@link DemandDepositAccountService#openDemandDepositAccount()}. The following is the algorithm for the method: <ol style="list-style-type: none"> <li>Instantiate {@link DDACloningBusinessPolicy} using the parameterized constructor by passing the existing DDA as the parameter.</li> <li>Validate the existing DDA by calling {@link DDACloningBusinessPolicy#validate()}. </li> <li>In case of any validation errors, throw an exception and stop processing.</li> <li>Call {@link DemandDepositAccountService#openDemandDepositAccount()} by passing {@link IDemandDepositAccount}. This will open the account.</li> <li>Since the account is opened independently, the balances for this new account are initialized to zero.</li> </ol> The following are the parameters taken by the method along with their significance <ul style="list-style-type: none"> <li>@param domainObject : The domain object for the demand deposit account which is to be opened. This is a clone of the original DDA. See {@link DemandDepositAccount} for more information.</li> </ul> </pre> Product level banking facilities will be applicable to the cloned. Note: Account level banking

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>facility is not copied to the cloned account. There might be a case when 1) product level facility is changed after the parent account is opened. 2) It might be required to configure banking facility explicitly for the cloned account. Maximum number of accounts that a party can have under the product as the primary (first) holder will be defined at the Product level. System should not permit CASA account opening if this account is breaching the set limit. System should permit CASA account opening only if the product code is active and not expired. It works as follows: &lt;li&gt;Get the count of account for which the party is primary account holder by calling <code>IDDAPartyAdapter#fetchCountPrimaryAccountHolder(partyId : String)</code>&lt;/li&gt; &lt;li&gt;Compare the count with the allowable count at the product level.&lt;/li&gt; &lt;li&gt;Reject account opening if the product level count is less than or equal to the count for which party is primary account holder. @param clonedAccount @param partyToAccountRelationship @param isDocumentIncomplete @param interestTicketNumber @throws FatalException</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllHoldersOfAccount		This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method validates the stop mail flag. If default communication addressType of account address linkage is same as party AddressType and if flag for stop mail is true then cheque book request cannot be made. @param account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	<p>This method fetches the address linkages for a particular party associated with the account. In the Inquiry DTO the docTypeId field would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiry Response. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the inputs. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6) Check for the validity of the party and its account relationship by calling fetchPartyAccountRelationship() method exposed by the PartyAccountRelationshipApplicationService with accountId and partyId as the input. If the party is not valid, "Invalid Party" error message is displayed. If the party does not have any relationship with the account, "No Relationship with the Account" error message is displayed. 7) Create an instance of the Domain Service (AccountAddressLinkageService). 8) Call the fetchPartyAddressLinkage() method of the AccountAddressLinkageService with accountId and</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>partyId as the input. 9) Receive the AccountAddressLinkage entity in response. 10) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) Call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) Call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO @param sessionContext</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method will fetch the address related to party and the address type. @param sessionContext @param accountId @param partyId @param postalAddressType @return DocumentAddressLinkageInquiryResponse @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method will fetch the postal address of the party for the current address type. @param sessionContext @param accountId @param partyId @param addressType @return PostalAddress

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method will fetch the postal address of the party for the current address type. @param sessionContext @param accountId @param partyId @param addressType @return PostalAddress
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method is used to get the details of the party used for the report generation for the newly opened account. It populates the default address(Home address) of the party, the mobile contact details and the email Id @partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	Sets the postal address details. Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	This method calls fetchPartyAddress() and returns partyAddressDTO. @param partyId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyAddress	This method will fetch party address. @param accountId @param addressType @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchRelationshipForPartyByRelationshipCode	This method will fetch all the party to party relationship response for given party code and relationship code. @param sessionContext	This validation will do following: Check the minor account status of the account The issuance of chequebook to a party depends on the Minor Account Status @ link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	er		@param partyId @param relationshipCode @return	<p>com.ofss.fc.enumeration.dda.MinorAccountStatus}          .&lt;br /&gt; &lt;br /&gt; There are three possible statuses at account level.&lt;br /&gt; 1) Major&lt;br /&gt; 2) MinorWithoutRestrictions&lt;br /&gt; 3) MinorWithRestrictions&lt;br /&gt; 1)Major: It will allow to perform the cheque book request.&lt;br /&gt; &lt;br /&gt; 2)MinorWithoutRestrictions: &lt;br /&gt; There are two parameters at the bank policy level.&lt;br /&gt; a) relation required&lt;br /&gt; b) relation type&lt;br /&gt; &lt;br /&gt; When relation Flag = FALSE&lt;br /&gt; If the relation required flag is false then there is no restriction for this status and cheque book request will be honored.&lt;br /&gt; &lt;br /&gt; When relation Flag = TRUE&lt;br /&gt; Validate if the relation type defined at the bank policy level is present in the party account relationship for this account.&lt;br /&gt; &lt;br /&gt; If present then allow cheque book request.&lt;br /&gt; &lt;br /&gt; If account DOES NOT have relationship present then &lt;br /&gt; throw business validation excetion.&lt;br /&gt; &lt;br /&gt; For example relation required flag is TRUE and relationship defined is Guardian then this account should have a guardian relation ship maintained for the account.&lt;br /&gt; &lt;br /&gt; 3)MinorWithRestrictions: &lt;br /&gt; There are two separate paramters maintained at bank policy level for this status as well. The rest of the processing is exactly same as MinorWithoutRestrictions.&lt;br /&gt; &lt;br /&gt; The derivation of the Account Minor status is created at account opening itself and it is updated if a parties minor status changes in batch.&lt;br /&gt; &lt;br /&gt; &lt;br /&gt; Validation for auto reorder of chequebook.&lt;br /&gt; The same rules of chequebook request for minor status will also be applicable for reorder of chequebook.&lt;br /&gt; @param account</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchRelationshipForPartyByRelationshipCode	This method will fetch all the party to party relationship response for given party code and relationship code. @param sessionContext @param partyId @param relationshipCode @return	This method will fetch all the party to party relationship response for given party code and relationship code. @param sessionContext @param partyId @param relationshipCode @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchRelationshipForPartyByRelationshipCode	This method will fetch all the party to party relationship response for given party code and relationship code. @param sessionContext @param partyId @param relationshipCode @return	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationshipForAmendOD	When new party is added in amend OD workflow, party id is not yet added in database. We are using simulation party id in P2A relationship. This method changes P2A without validating party id. @param sessionContext @param accountRelationshipDTOArray @throws FatalException	While updating P2A relationship. Business policy checks for operating instructions are from linked parties only. In case of borrower change in validate mode. If we do not close operating instructions of non borrower call was failing.
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	changePartyAccountRelationshipForAmendOD	When new party is added in amend OD workflow, party id is not yet added in database. We are using simulation party id in P2A relationship. This method changes P2A without validating party id. @param sessionContext @param accountRelationshipDTOArray @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method is used to get the details of the party used for the report generation for the newly opened account It populates the default address(Home address) of the party, the mobile contact details and the email Id @partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This method is used to fetch the party status whether it is Major,Minor or Operational_Major The different status combination possible are 1. Minor in case if both operational major and legal major is false 2. Operational major : In case if operational major is true and legal major is false 3. Legal major : In case if legal major is true and operational major is false If both legal and operation major is true it is an invalid case. @param partyId
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	This private method will return the {@link PartyType} for the partyId passed. @param sessionContext @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException	Returns the basic details and demographic details form the part of response. @param sessionContext @param partyId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method validates all the party id's are linked to the account.   <b>Since:</b> R2.2   @param accountAddressLinkageDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This will create a borrowing entity id.And stamp it on account. @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationShip that exists @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationShip that exists @param accountId @return
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input to This method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to check is warning indicator is present on party @param warningIndicatorTypeForRemoval @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input to This method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account, it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input to This method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This service will fetch a list of demand deposit accounts that are eligible for Overdraft facility.  An existing demand deposit account can be used only if the following conditions are fulfilled:     Parties applying for Overdraft facility on existing CASA accounts must be owners of the account either as Sole Owners or as Joint Owners.     <b>For e.g.</b> : If party A and party B apply for OD, then all CASA owned by Party A and Party B where each of them is Sole Owner and all CASA owned by both parties in joint are applicable for OD facility. If Party A and Party C hold a CASA, the same will not be applicable since Party C is not in the applicant list.     Also, existing CASA selected, must not be linked to any other Overdraft facility.     Refactored to this module from {@link LendingApplicationService} @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String[] partyId, Indicates the list of parties for which accounts are being fetched. @fcb.param MI, String[] offerId, Indicates the list of offer for which accounts are being fetched. @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ExistingCASAResponse, Indicates the accounts matching the given criteria. @throws FatalException @exception   {@link DemandDepositAccountErrorConstant#DDA_OFFER_INVALID} : FC_OR_DE_0081   This error occurs when:  The offer details are being fetched for incorrect offer identifier or offer has expired.   
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch all the parties of the account,it returns the party account relationship @param accountId @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	<p>TODO: Add order by partyId in the HQL fetchAllPartiesOfAccount. This method will be used to validate the accounts of the group. The accounts should satisfy the following conditions to be a part of the group. &lt;br&gt; &lt;ol&gt; &lt;li&gt;Should be a valid CASA Account.&lt;/li&gt; &lt;li&gt;Account Status should be valid for Group Account Maintenance.&lt;/li&gt; &lt;li&gt;All the accounts of the group should have the same currency.&lt;/li&gt; &lt;li&gt;The customer account relationship should be valid for Group Account Maintenance.&lt;/li&gt; &lt;li&gt;The account should not belong to more that one group with the same purpose.&lt;/li&gt; &lt;li&gt;There should be a minimum of 2 accounts involved in a group.&lt;/li&gt; The relationship between the account and the party should be in the role of a Customer and should be one of the following: &lt;ol&gt; &lt;li&gt;Sole Owner&lt;/li&gt; &lt;li&gt;Joint And First&lt;/li&gt; &lt;li&gt;Joint And Other&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;All the accounts of the group should be belong to the same party or set of parties as the case maybe.&lt;/li&gt; &lt;/ol&gt; For e.g. &lt;br&gt; &lt;table border=1&gt; &lt;tr&gt; &lt;td&gt;&lt;b&gt;AccountNumber&lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;PartiesToTheAccount&lt;/b&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;111&lt;/td&gt; &lt;td&gt;A~B&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;222&lt;/td&gt; &lt;td&gt;A~B&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;333&lt;/td&gt; &lt;td&gt;A~C&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;444&lt;/td&gt; &lt;td&gt;A~B&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;555&lt;/td&gt; &lt;td&gt;A~C&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;666&lt;/td&gt; &lt;td&gt;A&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; &lt;br&gt; In the above case: &lt;ol&gt; &lt;li&gt;Accounts 111, 222 and 444 can belong to the same group.&lt;/li&gt; &lt;li&gt;Accounts 333 and 555 can belong to the same group.&lt;/li&gt; &lt;li&gt;Accounts 111,222, and 555 cannot belong to the same group.&lt;/li&gt; &lt;li&gt;Accounts 111 and 666 cannot belong to the same group.&lt;/li&gt; &lt;/ol&gt;The method works as follows: &lt;ul&gt; &lt;li&gt;Declare</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>a local variable previousListOfParties = null.&lt;/li&gt;  &lt;li&gt;Declare a local variable previousCurrency = null.&lt;/li&gt; &lt;li&gt;Check the size of the array AccountGroupMember in the business policy DTO. If the size &lt; 2, then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Open a loop for the accounts of maintained for the group. &lt;li&gt;Load the account by calling repository method {@link DemandDepositAccountRepository#load} to validate if the account exists or not. If the account does not exist, then add an error message to the list of validation errors. &lt;/li&gt; &lt;li&gt;If the (previousCurrency &lt;&gt; null &amp;&amp; DemandDepositAccount.getCurrencyCode() &lt;&gt; previousCurrency), then add error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Set the value of previousCurrency = DemandDepositAccount.getCurrencyCode()&lt;/li&gt; &lt;li&gt;Check the status of the account by calling the service {@link DemandDepositAccount#validateStatus} to check if the status is valid for Group Account Maintenance. If the status is not valid, add error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Get the parties associated with the account along with their relationships by calling the service {@link IDDAPartyAdapter#fetchAllPartiesOfAccount}.&lt;/li&gt; &lt;li&gt;Open a loop for the response returned by the above service. &lt;ol&gt; &lt;li&gt;If the looprecord.partyId = accountGroup.partyId, then &lt;ol&gt; &lt;li&gt;If the looprecord.relationshipType is not one of the relationships mentioned above, then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;If the looprecord.roleId &lt;&gt; Customer, then add appropriate error message to the list of validation</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>errors.&lt;/li&gt; &lt;/ol&gt; &lt;li&gt;Concatenate the partyId in a tilde separated list and assign to variable listOfParties.&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;If none of the parties in the list = accountGroup.partyId, then add appropriate error message to the list ofValidation errors.&lt;/li&gt; &lt;li&gt;If (previousListOfParties != Null &amp;&amp; previousListOfParties &lt;&gt; listOfParties), then add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;li&gt;Set previousListOfParties = listOfParties.&lt;/li&gt; &lt;li&gt;Call the repository method {@link AccountGroupRepository#fetchNumberOfGroupsFor APurpose} to get the number of groups of the same purpose of which the account is a member. If the returned value &gt; 0, the add appropriate error message to the list of validation errors.&lt;/li&gt; &lt;/li&gt; &lt;/ul&gt;</p>
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method is used to store the details of all the party Id including the primary partyId @param partyToAccountRelationship @throws FatalException
IDDAPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the accountId  Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException	This method will be used to populate the DTO for activity log in case of modify account status is changed. @param accountId @param object @param sessionContext @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccountType		This method takes an accountId and accountType, and fetches basic account and associate primary owner's details @param sessionContext @param accountId @param accountType @return @throws FatalException
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountAddressLinkage		This method removes the account address linkage for the party who is deleted from {@link PartyToAccountRelationship} @param partyToAccountRelationshipArray
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountAddressLinkage		
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountAddressLinkage		
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccountType	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType) </pre>	Public method to search for a party. If party id is given then search only on the basis of party id Else Party type is a mandatory input Search for all search conditions @param sessionContext @param partySearchRequest @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccount	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType) </pre>	
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccount	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType) </pre>	
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccount	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType) </pre>	The following method is used to fetch the nickname details using the party id @param sessionContext @param partyId @param nickName @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccount	<p>/ (non-Javadoc)</p> <p>@seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService</p> <p>#</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)</p>	
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	fetchAccountByAccount	<p>/ (non-Javadoc)</p> <p>@seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService</p> <p>#</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)</p>	This method takes an accountId and accountType, and fetches basic account and associate primary owner's details @param sessionContext @param accountId @param accountType @return @throws FatalException
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	modifyAccountTitles		
IPartyAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAccountAdapter	modifyAccountTitles		This method is used to update the title for the accounts under a party id, while updating the party names. This method calls the {@link AccountApplicationService#modifyAccountTitles} through the {@link IPartyAccountAdapter} to update the account titles as well. @param sessionContext @param titleModificationRequestDTO @throws FatalException





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchInterestRatesForTD	Refer to {@link ContractApplicationService#getRegularInterestIndexOffering(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException	Refer to {@link ContractApplicationService#getRegularInterestIndexOffering(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchInterestRatesForTD	Refer to {@link ContractApplicationService#getRegularInterestIndexOffering(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception   {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchMarginsForTD	Refer to {@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException	Refer to {@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchMarginsForTD	Refer to {@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)} @param sessionContext @param	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>effectiveDate @param  pricingTicketNumber @param  productCode @param  productGroupId @param  originationFactsDTO @throws  FatalException</p>	<p>number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. &lt;br&gt; This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI, TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI, ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when: &lt;br&gt; Interest rate offerings cannot be fetched from Pricing module. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchServicingFeesForTD	@param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException	Refer to {@link ContractApplicationService#getPriceOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date, String[])}
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchServicingFeesForTD	@param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application</p> <p>@fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit.</p> <p>@throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	<p>@param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	@param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module.  This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	@param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param</p> <p>MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param</p> <p>MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param</p> <p>MI,TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	<p>@param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException</p>	<p>This method is used to payin deposit.&lt;br&gt; Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceForEvent	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getPriceForEvent(com.ofss.fc.app.adapter.td.dto.pricing.PriceRequestDTO)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceForEvent	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event	This method is used to fetch pricing ticket from pricing for original admin fee. Method will be called after the inquiry, so it will return persisted pricing ticket. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId term deposit account id @param depositId term deposit number @param redemptionType type of redemption @param redemptionDate date on which redemption performed @return {@link PricingResponse} Object containing pricing ticket @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceForEvent	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event	This method interacts with pricing module to fetch admin fees @param deposit Term deposit details @param redemptionDate Date on which redemption done @param eventId event constant string depending on redemption type @return PriceResponseDTO : OBP22IUT-17356 : since the pricing ticket is also required to be propagated, the response from pricing will be directly sent out to consume both admin fee amount and the pricing ticket for the pricing interaction
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceForEvent	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event	This method fetched Economic cost from pricing module. @param deposit term deposit details @param redemptionAmount amount to be redeemed @param redemptionDate date on which redemption done @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	createPricingPlan	<p>This method will create a pricing plan within pricing module, by calling a function to link&lt;br&gt; the arrangement number to the recommendation created earlier on the screen.&lt;br&gt; The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter&lt;br&gt; The implementation will handle the case where the ticket number is null by itself &lt;br&gt; calling for a recommendation from pricing and thus getting a ticket number.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date</p>	<p>This method will create a pricing plan within pricing module, by calling a function to link&lt;br&gt; the arrangement number to the recommendation created earlier on the screen.&lt;br&gt; The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter&lt;br&gt; The implementation will handle the case where the ticket number is null by itself &lt;br&gt; calling for a recommendation from pricing and thus getting a ticket number.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#createPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	createPricingPlan	<p>This method will create a pricing plan within pricing module, by calling a function to link the arrangement number to the recommendation created earlier on the screen. The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter. The implementation will handle the case where the ticket number is null by itself calling for a recommendation from pricing and thus getting a ticket number.</p> <p>@param depositFactsDTO {link TermDepositFactsDTO} Object containing facts required to create pricing plan  @param ticketNumber The pricing ticket required for pricing plan  @param effectiveDate The date which is either deposit or interest start date</p>	<p>Re-prices the complete {link PricingPlan} associated with the {code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. &lt;p&gt; The {link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {link PricingTicketDTO} before calling this method. The {link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {link FatalException} and aborts. &lt;/p&gt; @param sessionContext The session context @param depositFactsDTO {link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered. @param arrangementNumber The arrangement number @param effectiveDate The date on which to reprice @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	createPricingPlan	This method will create a pricing plan within pricing module, by calling a function to link  the arrangement number to the recommendation created earlier on the screen.  The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter  The implementation will handle the case where the ticket number is null by itself   calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	createPricingPlan	<p>This method will create a pricing plan within pricing module, by calling a function to link the arrangement number to the recommendation created earlier on the screen. The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter. The implementation will handle the case where the ticket number is null by itself calling for a recommendation from pricing and thus getting a ticket number.</p> <p>@param depositFactsDTO {link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date</p>	<p>This method is used to payin deposit. Method will be called from screen. @param payinTransactionDTO {link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {link PayinResponse} object containing outcome of Payin Transaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	createPricingPlan	This method will create a pricing plan within pricing module, by calling a function to link the arrangement number to the recommendation created earlier on the screen. The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter. The implementation will handle the case where the ticket number is null by itself calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date	This method will create a pricing plan within pricing module, by calling a function to link the arrangement number to the recommendation created earlier on the screen. The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter. The implementation will handle the case where the ticket number is null by itself calling for a recommendation from pricing and thus getting a ticket number. @param deposit {link Deposit} contains fields to fill the DTO sent to pricing module to create pricing plan. @param ticketNumber will identify the created recommendation.
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getFactsForPricingPolicyChart	This method will fetch details of facts required for evaluation of pricing policy chart. @param policyType Type of policy @param partyId CustomerId @param productGroupId TD product group Id @return FactDetailsDTO[] Array of fact details for given Pricing Policy chart	This method will fetch details of facts required for evaluation of pricing policy chart. @param policyType Type of policy @param partyId customerId @param productGroupId TD product group Id @return FactDetailsDTO[] Array of fact details for given Pricing Policy chart @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getFactsForPricingPolicyChart(com.ofss.fc.enumeration.pricing.PolicyChartLevel, com.ofss.fc.enumeration.pricing.PolicyType, java.lang.String, java.lang.String, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getFactsForPricingPolicyChart	This method will fetch details of facts required for evaluation of pricing policy chart. @param policyType Type of policy @param partyId CustomerId @param productGroupId TD product group Id @return FactDetailsDTO[] Array of fact details for given Pricing Policy chart	This method fetches fact details for PricePolicyChart. @param policyChartLevel {@link PolicyChartLevel} Policy chart level @param policyType {@link PolicyType} Type of policy @param customerId Id of party @param productGroupId Id of product group @param evaluateForDate Date for evaluation of PricePolicyChart @return Set of {@link FactDetails} objects
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtPricingPlan	This method will evaluate appropriate customer, product or bank level price policy chart(s) to derive TD interest rate. @param TermDepositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return {@link InterestDetailsDTO} Details of interest rate	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtPricingPlan	This method will evaluate appropriate customer, product or bank level price policy chart(s) to derive TD interest rate. @param TermDepositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return {@link InterestDetailsDTO}	This method gives the call to pricing and fetches the net Interest rate @param deposit {@link Deposit} Term deposit details @param endDate Contract modification value date @param startDate Contract modification date @param interestPayoutFrequency Interest payout frequency @param principalAmount Principal balance of deposit @return netInterestRate Interest rate applicable for term deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Details of interest rate	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtPricingPlan	This method will evaluate appropriate customer, product or bank level price policy chart(s) to derive TD interest rate. @param TermDepositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return {@link InterestDetailsDTO} Details of interest rate	This method do the inquiry of redemption amount. @param deposit Term deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money, com.ofss.fc.datatype.Date)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>This method will create a pricing plan within pricing module, by calling a function to link the arrangement number to the recommendation created earlier on the screen. The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter The implementation will handle the case where the ticket number is null by itself calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#createPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return {@link RateOfferingResponse} Object containing pricing ticket number @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRateOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown. The aim of this method is to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param effectiveDate The date which is either deposit or interest start date</p> <p>@param ticketNumber The pricing ticket number</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>Re-prices the complete {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained.</p> <p>&lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts.</p> <p>&lt;/p&gt; @param sessionContext The session context</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered.</p> <p>@param arrangementNumber The arrangement number</p> <p>@param effectiveDate The date on which to reprice</p> <p>@see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>This method fetches the Interest Rate for the given product from pricing module. @param sessionContext @param simulationParametersDTO @param termDepositFactsDTO @return interestRateFromPricing @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param effectiveDate The date which is either deposit or interest start date</p> <p>@param ticketNumber The pricing ticket number</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>This method will return pricing ticket. @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing input required to get pricing ticket</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param effectiveDate The date which is either deposit or interest start date</p> <p>@param ticketNumber The pricing ticket number</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>This method retrieves pricingTicket from pricing.</p> <p>@param sessionContext This parameter captures the basic information about the user of the system.</p> <p>@param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies.</p> <p>@return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket</p> <p>@throws FatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing, and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown</p> <p>The aim of this method if to begin a dialogue with pricing, the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param effectiveDate The date which is either deposit or interest start date</p> <p>@param ticketNumber The pricing ticket number</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>The method will be called to get the pricing ticket.</p> <p>@param contractDetailComputationRequest {@link ContractDetailComputationRequest} object containing details required to frequency details for deposit</p> <p>@param eventRateVarianceDTOs {@link EventRateVarianceDTO}</p> <p>@return {@link RateOffersFromPricing} holds the result of calling getOffersFromPricing()</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRateOffering	<p>Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing, and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown. The aim of this method is to begin a dialogue with pricing, the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here.</p> <p>@param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param effectiveDate The date which is either deposit or interest start date</p> <p>@param ticketNumber The pricing ticket number</p> <p>@return {@link RateOfferingResponse} Object containing pricing ticket number</p>	<p>This transaction will not be directly called from the screen.&lt;br&gt; Will be used by the reinvest transaction and external services can call this transaction.</p> <p>@param payinDTO {@link PayinRequestDTO} contains details of the fields need to open a deposit.</p> <p>@param componentPreferenceDTO {@link ComponentPreferenceDTO} contains details of the fields need to perform settlement.</p> <p>@param raiseAccountingFlag Flag indicating whether Accounting event needs to be raised or not</p> <p>@return {@link PayinResponse}</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param arrangementNumber - AccountId#DepositId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Array of {@link IndexFeatureDTO} objects containing different types of interest and margin charts	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param arrangementNumber - AccountId#DepositId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Array of {@link IndexFeatureDTO} objects containing different types of interest and margin charts	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO @param arrangementNumber - AccountId#DepositId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Interest Rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#fetchRatesForStageWithDateAndAmountFilter(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, java.lang.String, com.ofss.fc.datatype.Date, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param arrangementNumber - AccountId#DepositId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Array of {@link IndexFeatureDTO} objects containing different types of interest and margin charts	This method fetches the pricing details for the deposit @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to fetch interest rate from pricing module @param interestStartDate Interest start date of deposit @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param arrangementNumber - AccountId#DepositId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Array of {@link IndexFeatureDTO} objects containing different types of interest and margin charts	This method logs the Deposit Rate History. Whenever any Amount is Paid Out history will be maintained. Payouts will happen for the following events: <p><ol> <li>Interest Payout Instruction Execution</li> <li>Online Outstanding Interest Payout</li> <li>Interest Payout at Maturity</li> <li>Principal Payout at Maturity</li> <li>Principal Payout at Redemption</li> <li>Interest Payout at Redemption</li> </ol>The method will identify type of Transaction from {@link TransactionDetail#transactionType} and will create a instance of {@link PrincipalInterestPayoutHistory} which will be persisted. @param transactionDetail @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceOffering	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket	Refer to {@link ContractApplicationService#getPriceOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date, String[])} And  {@link ContractApplicationService#getRewardOfferingWithoutEvaluation()} @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceOffering	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket	This method will create a pricing plan within pricing module, by calling a function to link  the arrangement number to the recommendation created earlier on the screen.  The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter  The implementation will handle the case where the ticket number is null by itself   calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#createPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceOffering	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getPriceOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceOffering	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket	Re-prices the complete {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. <p> The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. </p> @param sessionContext The session context @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered. @param arrangementNumber The arrangement number @param effectiveDate The date on which to reprice

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceOffering	This method calls pricing to retrieve admin fees and pricing details @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket	The method will be called to get the pricing ticket. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} object containing details required to frequency details for deposit @param eventRateVarianceDTOs {@link EventRateVarianceDTO} @return {@link RateOffersFromPricing} holds the result of calling getOffersFromPricing()
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToRecommendation	This method will be called before account or deposit is opened for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances	Refer to {@link ContractApplicationService#updateUDMInRecommendation(SessionContext, String, EventRateVarianceDTO)} And {@link ContractApplicationService#mergeRateVarianceToRecommendation(SessionContext, PricingContractContextDTO, EventRateVarianceDTO[])} This service will add the UDM for CASA or OD against the Credit or Overdraft stage. This service is called when broker gives rebate in commission and that benefit is to be propagated to the customer. @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToRecommendation	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#mergeRateVarianceToRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO[])
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToRecommendation	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances	This method adds rate variances to pricing plan if contract modification happens  @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information of deposit facts @param eventRateVarianceDTO {@link EventRateVarianceDTO} Objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#mergeRateVarianceToPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToRecommendation	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToRecommendation	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances	The method will be called to get the pricing ticket.  @param contractDetailComputationRequest {@link ContractDetailComputationRequest} object containing details required to frequency details for deposit @param eventRateVarianceDTOs {@link EventRateVarianceDTO} @return {@link RateOffersFromPricing} holds the result of calling getOffersFromPricing()

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getVariationsForArrangementForStage	<p>Returns an array of type {@link EventRateVarianceDTO}, containing the rates supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur</p> <p>@param sessionContext Holds the information of the party/customer</p> <p>@param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails</p> <p>@param tdStage String array of different stages of the domain category</p> <p>@return An array of type {@link EventRateVarianceDTO} objects containing details of rate variances</p> <p>@see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRatesWithLenientComputation(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.enumeration.td.TermDepositStage)</p>	<p>This method retrieves pricingTicket from pricing.</p> <p>@param sessionContext This parameter captures the basic information about the user of the system.</p> <p>@param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies.</p> <p>@return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket</p> <p>@throws FatalException {@link FatalException} holds the exception coming from the transaction, if any</p> <p>@see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getVariationsForArrangementForStage	<p>Returns an array of type <a href="#">{@link EventRateVarianceDTO}</a>, containing the rates supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur</p> <p><b>@param</b> sessionContext Holds the information of the party/customer</p> <p><b>@param</b> termDepositFactsDTO <a href="#">{@link TermDepositFactsDTO}</a> Object containing TD parameters like partyId, productId, FactDetails</p> <p><b>@param</b> tdStage String array of different stages of the domain category</p> <p><b>@return</b> An array of type <a href="#">{@link EventRateVarianceDTO}</a> objects containing details of rate variances</p> <p><b>@see</b> com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRatesWithLenientComputation(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.enumeration.td.TermDepositStage)</p>	<p>The method will be called to get the pricing ticket.</p> <p><b>@param</b> contractDetailComputationRequest <a href="#">{@link ContractDetailComputationRequest}</a> object containing details required to frequency details for deposit</p> <p><b>@param</b> eventRateVarianceDTOs <a href="#">{@link EventRateVarianceDTO}</a></p> <p><b>@return</b> <a href="#">{@link RateOffersFromPricing}</a> holds the result of calling getOffersFromPricing()</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method calculates net rate applicable using base rate and userDiscretionaryMargin. @param simulationParameters {@link SimulationParameters} Holds all the input details to get rate @param pricingTicket Ticket from pricing module to get rate recommendations @return {@link InterestDetails} Object containing interest rate details

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method calculates net rate applicable using base rate and userDiscretionaryMargin. @param simulationParameters {@link SimulationParameters} Holds all the input details to get rate @param pricingTicket Ticket from pricing module to get rate recommendations @return {@link InterestDetails} Object containing interest rate details
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method returns the net interest rate based on the information present in {@link TDAccountConfigurationDTO}. It populates the TD facts and pass it to the service getNetRateAtRecommendation in the TD PricingAdapter. This service returns the net interest rate. @param sessionContext @param tdAccountConfigurationDTO @param tdFactsDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method retrieves pricingTicket from pricing. @param sessionContext This parameter captures the basic information about the user of the system. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies. @return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket @throws FatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getNetRateAtRecommendation	This method will fetch net rate from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates	@param contractDetailComputationRequest {@link ContractDetailComputationRequest} object containing details required to frequency details for deposit @return the net interest rate details derived from pricing

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	repriceAll	<p>Re-prices the complete {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. &lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. &lt;/p&gt; @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered. @param ticketNumber The pricing ticket @param effectiveDate The date on which to reprice</p>	<p>Re-prices the complete {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. &lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. &lt;/p&gt; @param sessionContext The session context @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered. @param arrangementNumber The arrangement number @param effectiveDate The date on which to reprice @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO @param arrangementNumber - AccountId#DepoistId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Interest Rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#fetchRatesForStageWithDateAndAmountFilter(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, java.lang.String, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method fetches the simulated from pricing. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method generates a pricing ticket number @return Pricing ticket number @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#generatePricingTicket()
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.</p> <p>This service is refactored to this module from <a href="#">{@link TermDepositSimulationApplicationService}</a> on March 27</p> <p><b>@fcb.param MI, SessionContext, sessionContext, SessionContext</b> which must contain the session context information such as user id, branch, branch code, channel etc.</p> <p><b>@fcb.param MI, ApplicationParameterDTO, applicationParameterDTO, ,</b> Indicates the extracted application details required for making a call to the pricing module</p> <p><b>@fcb.param MI, TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, ,</b> Indicates the product group details for the account configuration</p> <p><b>@fcb.param MI, ApplicantDTO, primaryApplicantDTO, ,</b> Indicates the information that is linked to the primary applicant of the application</p> <p><b>@fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, ,</b> Indicates the information linked to the account configuration for which pricing information is being fetched.</p> <p><b>@return NetInterestRateResponse, ,</b> Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit.</p> <p><b>@throws FatalException @exception</b> <a href="#">{@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND}</a> : FC_TD_333</p> <p>This error occurs when: Interest rate offerings cannot be fetched from Pricing module.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method fetches the pricing details for the deposit @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to fetch interest rate from pricing module @param interestStartDate Interest start date of deposit @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method gives the call to pricing and fetches the net Interest rate @param deposit {@link Deposit} Term deposit details @param endDate Contract modification value date @param startDate Contract modification date @param interestPayoutFrequency Interest payout frequency @param principalAmount Principal balance of deposit @return netInterestRate Interest rate applicable for term deposit
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generatePricingTicket	This method generates a pricing ticket number @return Pricing ticket number	This method fetches the Post Maturity Rate applicable for the Deposit @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToPlan	This method adds rate variances to pricing plan if contract modification happens  @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information of deposit facts @param eventRateVarianceDTO {@link EventRateVarianceDTO} Objects containing details of rate variances	This method adds rate variances to pricing plan if contract modification happens  @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information of deposit facts @param eventRateVarianceDTO {@link EventRateVarianceDTO} Objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#mergeRateVarianceToPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToPlan	This method adds rate variances to pricing plan if contract modification happens  @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information of deposit facts @param eventRateVarianceDTO {@link EventRateVarianceDTO} Objects containing details of rate variances	This method updates the udm to the existing pricing plan @param deposit @param eventRateVarianceDTO
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	mergeRateVarianceToPlan	This method adds rate variances to pricing plan if contract modification happens  @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information of deposit facts @param eventRateVarianceDTO {@link EventRateVarianceDTO} Objects containing details of rate variances	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	repriceTDDetails	Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. @param sessionContext @param termDepositFactsDTO @param repriceDate	Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. @param sessionContext @param termDepositFactsDTO @param repriceDate
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	repriceTDDetails	Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. @param sessionContext @param termDepositFactsDTO @param repriceDate	This method uses TD Pricing Adapter for repricing the pricing details for the given deposit as the input which reprices the following data at pricing module: repricePriceOfferings @param sessionContext @param deposit @param repriceDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchPersistedRates	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)	This method is used to update the deposit contract.  The customer requests for modification of Deposit amount and /or Deposit term of   term deposit within the window period (14 days default, defined at Brand and product level)  after Deposit placement or roll-over. The interest pay-out frequency can also be modified subject to Product rules  In contract modification allowed changes are  <li>Change in principal amount</li>  <li>Change in pay-out frequency</li>   <li>change in term</li> @param deposit {@link Deposit} Deposit details @param contractChangeHistory {@link ContractChangeHistory} Object containing history of contract modification @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object containing details required for settlement @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing details of rate variance at particular event @param pricingTicket The pricing ticket @param transactionReferenceNumber the unique number for a given transaction. @param transactionSubsequenceNumber the unique number for a given transaction in pair with transactionReferenceNumber,  if particular transaction composed of two or more transactions or else transactionSubsequenceNumber will always be 1 @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchPersistedRates	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchPersistedRates	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)	This method is used to fetch the Read only field during contract modification. The read only fields are as follows:- <ul style="list-style-type: none"> <li>Base Interest</li> <li>Margin</li> <li>UDM</li> <li>Net Interest Rate</li> <li>Interest Compounding Frequency</li> <li>Maturity Amont</li> </ul> @param depositContractModificationDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchPersistedRates	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)	This method fetches the persisted date as per the interest Start Date @param deposit @return InterestDetailsDTO
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistRates	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistRates	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistRates	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistRates	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method will create a pricing plan within pricing module, by calling a function to link  the arrangement number to the recommendation created earlier on the screen.  The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter  The implementation will handle the case where the ticket number is null by itself   calling for a recommendation from pricing and thus getting a ticket number. @param deposit {@link Deposit} contains fields to fill the DTO sent to pricing module to create pricing plan. @param ticketNumber will identify the created recommendation.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistBackDatedInterestRates	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method Creates the Pricing Ticket and gets the new Interest Rates applicable after the contract modification @param interestStartDate Interest start date of deposit @param interestPayoutFrequency {@link Frequency} Interest payout frequency @param principalAmount Principal balance of term deposit @param maturityTerm {@link Duration} Deposit term @param eventRateVarianceDTO {@link EventRateVarianceDTO} Object containing rate variances of event @param pricingTicket The pricing ticket @param persistedFlag TODO @param elapsedRateFlag TODO @param deposit {@link Deposit} Term deposit details @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistBackDatedInterestRates	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method performs a redemption. @param deposit Term deposit details @param redemptionTransactionDTO Details of redemption transaction @return {@link RedemptionResponseDetailsDTO} Object containing details of redemption after actual redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#doRedemption(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.app.td.dto.transaction.redemption.RedemptionTransactionDTO)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	persistBackDatedInterestRates	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method fetches the Post Maturity Rate applicable for the Deposit @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	linkBundleBenefitsToArrangement	<p>If bundle benefit applicable flag is true for an account, this method calls pricing to fetch applicable bundle benefits for an account.&lt;br&gt; Fetched applicable variances will be linked to an arrangement.&lt;br&gt; If at the time of payin linking is happening, then benefit applicability will be ALL as newly opened deposit will get interest and fee benefits.&lt;br&gt; If bundle benefits get applicable after deposit opening, then benefit applicability will be FEES as active deposits under given account will get only fee benefits.&lt;br&gt; @param arrangementNumber account id and deposit id with '#' separator @param partyId party id @param accountCurrency account currency code @param offerId TD offer id @param bundleId bundle id which benefits account will avail @param bundleBenefitsApplicableFlag flag which states whether benefits are applicable or not @param benefitApplicationStatus benefit applicability whether its interest and fees or only fees @param asOnDate the date when bundle benefits will be applicable @param principalBalance principal balance of deposit</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	linkBundleBenefitsToArrangement	<p>If bundle benefit applicable flag is true for an account, this method calls pricing to fetch applicable bundle benefits for an account.&lt;br&gt; Fetched applicable variances will be linked to an arrangement.&lt;br&gt; If at the time of payin linking is happening, then benefit applicability will be ALL as newly opened deposit will get interest and fee benefits.&lt;br&gt; If bundle benefits get applicable after deposit opening, then benefit applicability will be FEES as active deposits under given account will get only fee benefits.&lt;br&gt; @param arrangementNumber account id and deposit id with '#' separator @param partyId party id @param accountCurrency account currency code @param offerId TD offer id @param bundleId bundle id which benefits account will avail @param bundleBenefitsApplicableFlag flag which states whether benefits are applicable or not @param benefitApplicationStatus benefit applicability whether its interest and fees or only fees @param asOnDate the date when bundle benefits will be applicable @param principalBalance principal balance of deposit</p>	<p>Bundling service will call this method from poller or if someone want to refresh the bundle linkage of an account.&lt;br&gt;This method updates the account for bundle benefits whether it is applicable or not.&lt;br&gt; If bundle benefits are applicable for the account, the arrangement for all active deposit of an account will get linked to bundle benefits.&lt;br&gt; In this case, deposits can avail only fee benefits.&lt;br&gt; @param accountId term deposit account id @param isBundleBenefitApplicable boolean flag which states whether bundle benefits are applicable or not @param bundleId offer bundle id @param bundleReferenceId bundle reference id @param bundleEffectiveEndDate date which specifies the till when bundle benefits are effective for the account</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	linkBundleBenefitsToArrangement	<p>If bundle benefit applicable flag is true for an account, this method calls pricing to fetch applicable bundle benefits for an account.&lt;br&gt; Fetched applicable variances will be linked to an arrangement.&lt;br&gt; If at the time of payin linking is happening, then benefit applicability will be ALL as newly opened deposit will get interest and fee benefits.&lt;br&gt; If bundle benefits get applicable after deposit opening, then benefit applicability will be FEES as active deposits under given account will get only fee benefits.&lt;br&gt; @param arrangementNumber account id and deposit id with '#' separator @param partyId party id @param accountCurrency account currency code @param offerId TD offer id @param bundleId bundle id which benefits account will avail @param bundleBenefitsApplicableFlag flag which states whether benefits are applicable or not @param benefitApplicationStatus benefit applicability whether its interest and fees or only fees @param asOnDate the date when bundle benefits will be applicable @param principalBal principal balance of deposit</p>	<p>This method is used to payin deposit.&lt;br&gt; Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	updateBundleBenefitApplicationStatus	@param arrangementNumber account id and deposit id with '#' separator @param asOnDate the date when bundle benefits will be applicable @param fixedBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees @param variableBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees	This method will update BundleBenefitApplicationStatus in pricing depending on the flag isBundleBenefitApplicable. @param arrangementNumber @param benefitApplicabilityDTOArray
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	updateBundleBenefitApplicationStatus	@param arrangementNumber account id and deposit id with '#' separator @param asOnDate the date when bundle benefits will be applicable @param fixedBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees @param variableBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	updateBundleBenefitApplicationStatus	@param arrangementNumber account id and deposit id with '#' separator @param asOnDate the date when bundle benefits will be applicable @param fixedBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees @param variableBenefitAppplicationStatus benefit applicability whether its	@param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interest and fees or only fees	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	updateBundleBenefitApplicationStatus	@param arrangementNumber account id and deposit id with '#' separator @param asOnDate the date when bundle benefits will be applicable @param fixedBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees @param variableBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees	@param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	updateBundleBenefitApplicationStatus	@param arrangementNumber account id and deposit id with '#' separator @param asOnDate the date when bundle benefits will be applicable @param fixedBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees @param variableBenefitAppplicationStatus benefit applicability whether its interest and fees or only fees	@param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchNetRatePlanForArrangementNumber	This method will return net rate plan which consists of array of net rates existing for the given arrangement number which will be used in persisting transaction state for reversal of redemption @param arrangementNumber @return NetRatePlanDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchNetRatePlanForArrangementNumber	This method will return net rate plan which consists of array of net rates existing for the given arrangement number which will be used in persisting transaction state for reversal of redemption @param arrangementNumber @return NetRatePlanDTO	The method saves the transaction state of a transaction.  This state would be maintained if reversal id performed on the transaction. @param accountId This attribute denotes a label representing one or more numbers of codes by which an account may be uniquely identified. These identifiers may be industry recognized identifiers or internally generated identifiers. The identifier is generated when the account relationship is first established and it never undergoes a change after that. @param depositId This deposit number represents a unique deposit in the account.  <p> The deposit number is incremented every time a new deposit is created in an account. It may or may not be created due to interest payout or renewal.  @param transactionReferenceNumber Holds the transactionReferenceNumber of the current transaction. @param sequenceNumber Holds the transactionSequenceNumber of the current transaction. @param tdTransactionType Holds the value to identify the type of transaction. @param isTransactionReversalPossible Will tell if the reversal is possible for the current transaction. @param taskCode Hold task code of the current transaction. @return {@link TransactionState}
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	restoreNetRatesFromPricing	This method will restore the net rate plan for reversal of the following events <ul> <li>Premature part redemption <li>Contract Modification </ul> Hence if re-pricing has happened at the time of the above mentioned events, it will get reversed. @param netRatePlanDTO @param effectiveDate	This method calls pricing module to restore the interest rate which has been changed in pricing due to the following events <ul> <li>Premature part redemption <li>Contract Modification </ul> @param netRatePlanDTO {@link NetRatePlanDTO} Object containing net rate details for the arrangement number

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceFeatureDTOForEvent	This method will fetch price feature for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceFeatureDTO} Information about price for given Term Deposit Event	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceFeatureDTOForEvent	This method will fetch price feature for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceFeatureDTO} Information about price for given Term Deposit Event	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	This method adds event rate variance for agent UDM to recommendation. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing deposit facts @param valueDate start date from when udm will be applicable @param pricingTicket pricing ticket generated @param offerId Term deposit offer id @param productGroupCode product group code of term deposit account @param termDeposit {@link Deposit} Object containing deposit details @param maturityDate end date till when udm will be applicable @param rebateBenefitMargin user discretionary margin from agent	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	This method adds event rate variance for agent UDM to recommendation. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing deposit facts @param valueDate start date from when udm will be applicable @param pricingTicket pricing ticket generated @param offerId Term deposit offer id @param productGroupCode product group code of term deposit account @param termDeposit {@link Deposit} Object containing deposit details @param maturityDate end date till when udm will be applicable @param rebateBenefitMargin user discretionary margin from agent	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module.  This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @fcb.param MI, String, partyId, Indicates the party identifier. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	This method adds event rate variance for agent UDM to recommendation. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing deposit facts @param valueDate start date from when udm will be applicable @param pricingTicket pricing ticket generated @param offerId Term deposit offer id @param productGroupCode product group code of term deposit account @param termDeposit {@link Deposit} Object containing deposit	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the term deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number. This service is refactored to this module from {@link TermDepositSimulationApplicationService} on March 27 @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>details @param maturityDate end date till when udm will be applicable @param rebateBenefitMargin user discretionary margin from agent</p>	<p>context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,TermDepositProductGroupLinkageDTO, termDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, TDAccountConfigurationDTO, tdAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration, net interest rate and maturity date for the term deposit. @throws FatalException @exception &lt;br&gt; {@link TermDepositErrorConstants#INTEREST_OFFERINGS_NOT_FOUND} : FC_TD_333 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	addUDMtoRecommendation	This method adds event rate variance for agent UDM to recommendation. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing deposit facts @param valueDate start date from when udm will be applicable @param pricingTicket pricing ticket generated @param offerId Term deposit offer id @param productGroupCode product group code of term deposit account @param termDeposit {@link Deposit} Object containing deposit details @param maturityDate end date till when udm will be applicable @param rebateBenefitMargin user discretionary margin from agent	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricng based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method will fetch the stored rate as of evaluationdate @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO @param arrangementNumber - AccountId#DepoistId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Interest Rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#fetchRatesForStageWithDateAndAmountFilter(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, java.lang.String, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method will inquire the rates from pricing as of current posting date based on the status of the deposit, i.e. The stage for which the rates are fetched will depend on the deposit status.   For example : <ul> <li>For Deposit Status : Matured, Unclaimed : TermDepositStage will be set to PMI </li>For other Deposit Status : TermDepositStage will be set to REGULAR </ul> @param deposit @param stage TODO @param depositAttributeDTO @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method will fetch the rates from pricing recommendation for a particular stage. @param arrangementNumber @param stage @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method will return the net rate of the Deposit as on the interest start date. @param deposit @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method fetches the pricing details for the deposit @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to fetch interest rate from pricing module @param interestStartDate Interest start date of deposit @return Array of {@link IndexFeatureDTO} objects containing interest rates applicable to deposit
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchRatesForStage	This method will fetch the rates from pricing based on the arrangement number and stage. @param arrangementNumber @param stage @param evaluationDate @return	This method logs the Deposit Rate History. Whenever any Amount is Paid Out history will be maintained. Payouts will happen for the following events: <p><ol> <li>Interest Payout Instruction Execution</li><li>Online Outstanding Interest Payout</li><li>Interest Payout at Maturity</li> <li>Principal Payout at Maturity</li> <li>Principal Payout at Redemption</li> <li>Interest Payout at Redemption</li> </ol>The method will identify type of Transaction from {@link TransactionDetail#transactionType} and will create a instance of {@link PrincipalInterestPayoutHistory} which will be persisted. @param transactionDetail @throws FatalException
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generateUserDiscretionaryMarginIdForPrice	function to generate udm id for price @param ticketNumberOrArrangementNumber @param eventCode @param priceCode @param derivationClauseId @return	function to generate udm id for price @param ticketNumberOrArrangementNumber @param eventCode @param priceCode @param derivationClauseId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	generateUserDiscretionaryMarginIdForPrice	function to generate udm id for price @param ticketNumberOrArrangementNumber @param eventCode @param priceCode @param derivationClauseId @return	This method fetches the pricing details which are required to populate the fee negotiation work list item dto.  It fetches original admin fee and calculates the negotiated amount.  Also sets the price definition name, event name and arrangement number as account id.  @param accountId term deposit account id @param depositNo term deposit number @param redemptionType Type of redemption to be performed @param modifiedAdminFee Fees modified by OE @param pricingTicket TODO @return {@link FeeNegotiationApprovalWorkItemDTO}
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method will fetch price for given Term Deposit Event, from Pricing module. @param termDepositFactsDTO {@link TermDepositFactsDTO} object containing facts required to get price from pricing @param eventId The event id for which price needs to be fetched @param evaluationDate Evaluation date @param currencyInWhichPriceRequired The currencyCode @return {@link PriceResponseDTO} Information about price for given Term Deposit Event @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getPriceForEvent(com.ofss.fc.app.adapter.td.dto.pricing.PriceRequestDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method will create a pricing plan within pricing module, by calling a function to link  the arrangement number to the recommendation created earlier on the screen.  The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter  The implementation will handle the case where the ticket number is null by itself   calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#createPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getPriceOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate <a href="#">{@link BankingFactContextDTO}</a> from <a href="#">{@link TermDepositFactsDTO}</a> @param termDepositFactsDTO <a href="#">{@link TermDepositFactsDTO}</a> Using which <a href="#">{@link BankingFactContextDTO}</a> needs to be populated @return Populated <a href="#">{@link BankingFactContextDTO}</a> Object with banking facts	Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO <a href="#">{@link TermDepositFactsDTO}</a> Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return <a href="#">{@link RateOfferingResponse}</a> Object containing pricing ticket number @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRateOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate <a href="#">{@link BankingFactContextDTO}</a> from <a href="#">{@link TermDepositFactsDTO}</a> @param termDepositFactsDTO <a href="#">{@link TermDepositFactsDTO}</a> Using which <a href="#">{@link BankingFactContextDTO}</a> needs to be populated @return Populated <a href="#">{@link BankingFactContextDTO}</a> Object with banking facts	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method is called to fetch the Interest Rate when the Deposit Contract is modified @param depositFactsDTO @param arrangementNumber - AccountId#DepoistId @param stage - DepositStatus(Regular) @param evaluationDate - PostingDate @return Interest Rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#fetchRatesForStageWithDateAndAmountFilter(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, java.lang.String, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method fetches the simulated from pricing. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method is used to persist rate in pricing as of current date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method is used to persist rate in pricing as of passed date. @param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#mergeRateVarianceToRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO[])

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	Returns an array of type {@link EventRateVarianceDTO}, containing the rates supposed to be applied for all events defined for the given arrangement/account number. The returned array will contain one element per event. We will do lenient computation using this method, when exactly we don't have the all the facts like transactionAmount, i.e to show how the charges will apply to the events if they occur <p> @param sessionContext Holds the information of the party/customer @param termDepositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param tdStage String array of different stages of the domain category @return An array of type {@link EventRateVarianceDTO} objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRatesWithLenientComputation(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.enumeration.td.TermDepositStage)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate <a href="#">{@link BankingFactContextDTO}</a> from <a href="#">{@link TermDepositFactsDTO}</a> <a href="#">{@param termDepositFactsDTO}</a> <a href="#">{@link TermDepositFactsDTO}</a> Using which <a href="#">{@link BankingFactContextDTO}</a> needs to be populated <a href="#">@return</a> Populated <a href="#">{@link BankingFactContextDTO}</a> Object with banking facts	This method will fetch rates from pricing for Pricing recommendation created for deposit <a href="#">{@param termDepositFactsDTO}</a> <a href="#">{@link TermDepositFactsDTO}</a> Containing details of facts required for fetching net rates <a href="#">{@param evaluationDate}</a> Date for which rate needs to be fetched <a href="#">{@param pricingTicket}</a> Pricing ticket required for interaction with pricing module <a href="#">@return</a> <a href="#">{@link InterestDetailsDTO}</a> Containing information about interest rates <a href="#">@see</a> com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.o.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate <a href="#">{@link BankingFactContextDTO}</a> from <a href="#">{@link TermDepositFactsDTO}</a> <a href="#">{@param termDepositFactsDTO}</a> <a href="#">{@link TermDepositFactsDTO}</a> Using which <a href="#">{@link BankingFactContextDTO}</a> needs to be populated <a href="#">@return</a> Populated <a href="#">{@link BankingFactContextDTO}</a> Object with banking facts	Re-prices the complete <a href="#">{@link PricingPlan}</a> associated with the <a href="#">{@code arrangementNumber}</a> . This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. <p>&lt;p&gt; The <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)}</a> <a href="#">TicketService.startDialogue(SessionContext)}</a> method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling this method. The <a href="#">{@link PricingTicketDTO}</a> must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a <a href="#">{@link FatalException}</a> and aborts. &lt;/p&gt; <a href="#">{@param sessionContext}</a> The session context <a href="#">{@param depositFactsDTO}</a> <a href="#">{@link TermDepositFactsDTO}</a> Object containing information about deposit for which pricing plan is to be created registered. <a href="#">{@param arrangementNumber}</a> The arrangement number</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param effectiveDate The date on which to reprice @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. @param sessionContext @param termDepositFactsDTO @param repriceDate
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getBankingFactContextDTO	This method is used to populate {@link BankingFactContextDTO} from {@link TermDepositFactsDTO} @param termDepositFactsDTO {@link TermDepositFactsDTO} Using which {@link BankingFactContextDTO} needs to be populated @return Populated {@link BankingFactContextDTO} Object with banking facts	This method populates facts in {@link PricingContractContextDTO} for given arrangement number. @param arrangementNumber arrangement number from pricing @return {@link PricingContractContextDTO} DTO containing facts required by pricing module
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	getRatesAtRecommendationForStage	This method os used to get the rates from pricing for the given stage @param termDepositFactsDTO @param evaluationDate @param pricingTicket @param stage @return	@param contractDetailComputationRequest {@link ContractDetailComputationRequest} object containing details required to frequency details for deposit @return the net interest rate details derived from pricing
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchSimulatedRateFromPricing	@param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method will evaluate appropriate customer, product or bank level price policy chart(s) to derive TD interest rate. @param TermDepositFactsDTO TD parameters like partyId, productId, FactDetails @param evaluationDate Date for which rate needs to be fetched @return InterestDetailsDTO Details of interest rate @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO)
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPricingAdapter	fetchSimulatedRateFromPricing	@param depositFactsDTO @param arrangementNumber @param stage @param evaluationDate @param pricingTicket @return	This method will fetch the rates from pricing recommendation for a particular stage. @param arrangementNumber @param stage @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPricingAdapter	getPriceFeatureDTOForPricingTicket	Get the price information when the ticket number is available. @param termDepositFactsDTO @param pricingTicket @param currencyInWhichPriceRequired @return	This method fetches the pricing details which are required to populate the fee negotiation work list item dto.  It fetches original admin fee and calculates the negotiated amount.  Also sets the price definition name, event name and arrangement number as account id.  @param accountId term deposit account id @param depositNo term deposit number @param redemptionType Type of redemption to be performed @param modifiedAdminFee Fees modified by OE @param pricingTicket TODO @return {@link FeeNegotiationApprovalWorkItemDTO}



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	<p>This method executes the settlement as per the mode selected for the split amount. @param splitSettlementAmount Indicates the final settlement amount in the settlement currency to be settled via given mode @param messageType @param arrangementId @param priority @param isOnline @param narration @param splitTransactionAmount @param shortNarration @param additionalParameter @param offerId TODO @param exchangeRateForReverseCalculation TODO @param partyId TODO @param referenceDetailsDTO TODO @param mode Indicates the details of the mode to be used for settling this split amount return Returns the status of the execution for this mode @param acyTcyExchangeRate TODO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	executeSettlement	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param productCode @param eventCode @param moduleReferenceNo @param messageType @param arrangementId @param priority @param moduleCode @param isOnline @param narration @param shortNarration @param transactionLiteral @param additionalParameter @param offerId @param rateDetail @param partyId @param referenceDetailsDTO TODO @param amount The amount to be settled @param splitTransactionAmount @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the entire settlement mode details and not just account. @param sessionContext @param amount @param modeDetails @param direction @param transactionType @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input card details and determine tentative fees if any @param context @param amount @param details</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input cheque details and determine tentative fees if any (say for foreign cheque deposit) @param context @param amount @param details @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input collection details and determine tentative fees if any (for network)  @param context @param amount @param details  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input CASA account details. This method will not determine any fees @param context @param amount @param details @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input ledger details. This method will not determine any fees @param context @param amount @param details @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input MC issue details and determine tentative fees if any (for issuance)  @param context @param amount @param details  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input payments details and determine tentative fees if any (for network)  @param context @param amount @param details  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input term deposit details. This method will not determine any fees. If the money is sent, then it will check if the monetary value is within the deposit creation limits. @param context @param amount @param details @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input card details and determine tentative fees if any @param context @param amount @param details</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input card details and determine tentative fees if any @param context @param amount @param details</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method will validate the input loan details. This method will not determine any fees. If the money is sent, then it will check if the monetary value is within the deposit creation limits. @param context @param amount @param details @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	validateSettlementRequest	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayments, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will Independently keep track of the initiation date. @param context @param eventCode @param amount @param details @param direction @param transactionType @param callerModuleType @param partyId @param arrangementId</p>	<p>This method should be called at time of submission of a component preference or settlement preference to the repository. This will do basic validation on the inputs. The response will also indicate when should settlement engine initiate any execution on its own so that the funds can be made available on time. For eg, for a Loans PDC for arrear repayment, the settlement route will be cheque. When the clearing settlement mode is invoked, and the value date is sent to clearing API, then the API will return the proposed activation date, and the value date. This proposed activation date will be the initiation date. Although the caller module will keep track of the Arrear Appropriation Event as of value date, the settlement engine will independently keep track of the initiation date. The module adapter will have to determine the facts required by e-Pricing to determine mode usage charges. Eg, for payments, the network related charges can be calculated only if the network id as a fact is known @param context @param amount @param details @param direction @param transactionType @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction INDicates the direction of fund flow from Customer perspective @param transactionType INDicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	<p>This method is used to iterate the settlement reversal response..This method is called in case of reversal of an executed settlement instruction. @param settlementReversalResponseDTO @param i @param objects</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	<p>This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementModuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementModuleAdapter	reverseSettlementInstruction	This is the basic method for invocation of module API based on the settlement mode. @param settlementDirectionType @param narration @param subsequenceNumber @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException @throws FatalException	
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityToken		
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityToken		This method adds security token in the given MimeHeaders.  @fcb.param MI,MimeHeaders,mimeHeaders, ,Indicates instance of MimeHeaders. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityToken		This method adds security token in the given Map.  @fcb.param MI,Map,httpHeaders, ,Indicates instance of Map. 
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityTokenInWebServiceRequest	This method adds security token in the given MimeHeaders.  @fcb.param MI,MimeHeaders,mimeHeaders, ,Indicates instance of MimeHeaders. 	This method adds security token in the given MimeHeaders.  @fcb.param MI,MimeHeaders,mimeHeaders, ,Indicates instance of MimeHeaders. 
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityTokenInWebServiceRequest	This method adds security token in the given MimeHeaders.  @fcb.param MI,MimeHeaders,mimeHeaders, ,Indicates instance of MimeHeaders. 	This method adds security token in the given Map.  @fcb.param MI,Map,httpHeaders, ,Indicates instance of Map. 
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityTokenInWebServiceRequest	This method adds security token in the given Map.  @fcb.param MI,Map,httpHeaders, ,Indicates instance of Map. 	This method adds security token in the given MimeHeaders.  @fcb.param MI,MimeHeaders,mimeHeaders, ,Indicates instance of MimeHeaders. 
ITrustServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ITrustServiceAdapter	addSecurityTokenInWebServiceRequest	This method adds security token in the given Map.  @fcb.param MI,Map,httpHeaders, ,Indicates instance of Map. 	This method adds security token in the given Map.  @fcb.param MI,Map,httpHeaders, ,Indicates instance of Map. 
IHardshipReliefRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\IHardshipReliefRequestAdapter	updateHardshipReliefRequest	@param applicationId @param partyId @param hardshipReliefUpdataionType @param hardshipExtensionDate @return TransactionStatus @throws FatalException	@param applicationId @param partyId @param hardshipReliefUpdataionType @param hardshipExtensionDate @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IHardshipReliefRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IHardshipReliefRequestAdapter	invokeHardshipReliefProcess	@param hardshipReliefRequestDTO @return HardshipReliefRequestInquiryResponse @throws FatalException	@param hardshipReliefRequestDTO @return HardshipReliefRequestInquiryResponse @throws com.ofss.fc.infra.exception.FatalException
IHardshipReliefRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IHardshipReliefRequestAdapter	deleteWorkListItemForTransactionReference		@param applicationId @param partyId @throws FatalException
IHardshipReliefRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IHardshipReliefRequestAdapter	validateHardshipReliefRequest	@param hardshipReliefRequestDTO @return TransactionStatus @throws FatalException	@param hardshipReliefRequestDTO @return HardshipReliefRequestInquiryResponse @throws com.ofss.fc.infra.exception.FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacilities	This method fetches the brokers for an array of facility Id provided @param appContext @param facilityId @param agentArrangementType @return @throws FatalException	This method is used to stamp the broker to the facility when portfolio transfer is being done If it is exclusion scenario, then it fetches all the facility for a given Broker and then stamps the to broker and other agent details to it and adds to the DB. If it is inclusion scenario, then for each account it fetches the attached facility and adds the corresponding new Broker to the facility and adds it to DB. @param sessionContext @param agentBulkArrangementDetailsDTO @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacilities	This method fetches the brokers for an array of facility Id provided @param appContext @param facilityId @param agentArrangementType @return @throws FatalException	This method is called to fetch all the Brokers for a list of facilities. As input a list of facilities is provided and as output in the response array of broker DTO is returned which contains broker Details @param appContext @param facilityId @param agentArrangementType @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacilities	This method fetches the brokers for an array of facility Id provided @param appContext @param facilityId @param agentArrangementType @return @throws FatalException	This method is called to fetch all the Brokers for a list of accountIds. As input a list of accounts is provided and as output in the response array of broker DTO is returned which contains broker Details @param appContext @param facilityId @param agentArrangementType @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacilities	This method fetches the brokers for an array of facility Id provided @param appContext @param facilityId @param agentArrangementType @return @throws FatalException	This method will call the fetchBrokerForFacilities service of the Agent Arrangement Linkage Application service to query and return the broker Details for the array of facilities Id provided
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchTermDepositAccountDetails	This method fetches the TD account details for an array of accountIds @param sessionContext @param accountIds @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receipt Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchDepositDetails	This method fetches the deposit details for td account for a given account Id @param sessionContext @param accountId @return @throws FatalException	This method is used to fetch the CASA account Details based on the accountIds. In this method the accountIds(arrangementIds) are passed which is used to fetch the CASA Account Details.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchTermDepositSearchDetails	This method is called from ui to search and fetch all the accounts and their details which fall under the given search criteria @param sessionContext @param searchDTO @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from TD module and then agentArrangementLinkage DB is queried to get commission Receiptent Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchSubmissionDetails	This method is used to query and fetch submission details for an array of submission Ids @param sessionContext @param submissionId @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receiptent Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchSubmissionSearchDetails	This method is used to search the submission records based on the search DTO provided @param sessionContext @param searchDTO @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from ORIGINATION module and then agentArrangementLinkage DB is queried to get commission Receiptent Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchODDetails	This method is used to fetch OD account details for an array of account Ids @param sessionContext @param accountIds @param commissionRecipientIdMap @param commissionRecipientNameMap @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				agentArrangementLinkage DB the corresponding commission Recipient Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchODDetails	This method is used to fetch OD account details for an array of account Ids @param sessionContext @param accountIds @param commissionRecipientIdMap @param commissionRecipientNameMap @return @throws FatalException	This method fetches the OD Account Details and also populates the commission recipient ID and Name while returning the OD DTO
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchODDetails	This method is used to fetch OD account details for an array of account Ids @param sessionContext @param accountIds @param commissionRecipientIdMap @param commissionRecipientNameMap @return @throws FatalException	This method is called to fetch OD Account Details based pn the search Parameters provided. Search Parameters include account Id,from Date,to Date and Party Id.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchSearchODDetails	This method is used to fetch the OD account based on the search criteria provided @param sessionContext @param searchDTO @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from OD module and then agentArrangementLinkage DB is queried to get commission Recipient Id and Name @param sessionContext @param searchDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchLoanDetails	This method is used to fetch loan account details for an array of accountIds @param sessionContext @param accountIds @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receipt Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchSearchLoanDetails	This method is used to search and fetch the loan accounts and their details based on input criteria @param sessionContext @param searchDTO @return @throws FatalException	This method is used to fetch the Details which the user has queried for a particular module. First the queries submissinIds and details are fetched from LOAN module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacility	This method is used to fetch broker for a single facilityId @param sessionContext @param facilityId @return @throws FatalException	This method checks if for a facility Id provided the broker that is going to be stamped is already present in the DB against the same Facility Id. @param agentArrangementLinkageDTO @return
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacility	This method is used to fetch broker for a single facilityId @param sessionContext @param facilityId @return @throws FatalException	This method checks if for a facility Id provided the broker that is going to be stamped is already present in the DB against the same Facility Id. @param agentArrangementLinkageDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacility	This method is used to fetch broker for a single facilityId @param sessionContext @param facilityId @return @throws FatalException	This method is used to fetch the agent arrangement Details for a single facility ID. A facility can be linked to multiple accounts and can have multiple brokers , hence this method returns all the agent arrangements for the given facility. @param sessionContext @param facilityId @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchBrokerForFacility	This method is used to fetch broker for a single facilityId @param sessionContext @param facilityId @return @throws FatalException	This adapter may be called to fetch the broker details for a particular facility Id.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountDetailsForAgent	This method is used to fetch account details for an array of account ids for CASA @param sessionContext @param accountIds @param commissionRecipientIdMap @param commissionRecipientNameMap @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receptient Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountDetailsForAgent	This method is used to fetch account details for an array of account ids for CASA @param sessionContext @param accountIds @param commissionRecipientIdMap @param commissionRecipientNameMap @return @throws FatalException	This method is used to fetch the Account details such as Account id, Account title, Party id, Account Opening Date, Account Balance. This method fetches the account basic details from fetchAccountDetailsForAgent method. For each account id, it fetched the party ids from party account relationship service. For each account id it fetches the account balance from dda inquiry application service

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountDetailsForBrokerSearch	This method is used to search and fetch accounts for CASA based on input search parameters. @param sessionContext @param searchDTO @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from CASA module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountDetailsForBrokerSearch	This method is used to search and fetch accounts for CASA based on input search parameters. @param sessionContext @param searchDTO @return @throws FatalException	This method can be called to fetch Casa Account Details based on the search parameters provided. Search parameters include from date, to date,Account Id and Party Id.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountsCountForSingleCustomer	This method is used to fetch all accounts for a given partyid @param sessionContext @param partyId @return @throws FatalException	This method fetches the total count of the arrangements of a broker for all the modules which include CASA,OD,LOANS,TD and ORIGINATION. if fetchFromAgentDB is true then it fetches the arranhementIds from the agentArrangementLinkage DB for a given BrokerId Otherwise it calls each module to get the corresponding arrangementIds on the basis of party id and then checks from agentArrangementLinkage DB to get the count of only those arrangementIds which belong to the required broker. Finally the count of the arrangementIDs of each module is returned in the CountDTO.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchAccountsForSingleCustomer	This method is used to fetch all accounts of a given module and given party id @param sessionContext @param partyId @param taskModuleType @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				agentArrangementLinkage DB the corresponding commission Recipient Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	@param sessionContext @param arrangementId @param agentArrangementLinkageResponse @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Recipient Details are fetched.
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the agent arrangement Details for a single facility ID. A facility can be linked to multiple accounts and can have multiple brokers , hence this method returns all the agent arrangements for the given facility. @param sessionContext @param facilityId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from ORIGINATION module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from CASA module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from TD module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from LOAN module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to fetch the Details which the uses has queried for a particular module. First the queries submissinIds and details are fetched from OD module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @param searchDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchPartyName	This method is used to fetch the party name for a given commission recipient id. @param sessionContext @param commissionRecipientId @param agentType @return @throws FatalException	This method is used to get the list of valid arrangementIds from the list of arrangementIds passed passed to it. The input List of arrangement Ids which is received from different modules is validated from the agent arrangement linkage DB to check if it belongs to a given agentId. The resulting list of matched arrangement Ids is returned back. Apart from this the commission Recipient Name and Commission Recipient Id are populated. @param sessionContext @param list @param agentId @param taskModuleType @param commissionRecipientIdMap @param commissionRecipientNameMap @return
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchCommissionRecipientStatus	This method is used to fetch the status of Commission REcipient Id @param sessionContext @param commissionRecipientId @return @throws FatalException	This method is used to update the commission recipient's name and Id it fetches the agent record based on the arrangementId and agentId , then updates the commission recipient's name and Id it is used when the commision Recipient of a account is changed
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchCommissionRecipientAndPartyID	This method is used to fetch commission recipient and party id for a given agent. @param sessionContext @param agentId @param agentType @return @throws FatalException	@param sessionContext @param agentArrangementLinkageNew @throws FatalException
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchCommissionRecipientAndPartyID	This method is used to fetch commission recipient and party id for a given agent. @param sessionContext @param agentId @param agentType @return @throws FatalException	This method will check if the broker which is going to be added for a facility already exists in DB or not. If not then it takes the entity object changes the agentID,agent Party id and commission Recipient Id and send to add to DB. @param sessionContext @param agentBulkArrangementDetailsDTO @param agentArrangementLinkage @param facilityAddedForNewBroker @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	validateAccountStatus	This method is used to validate an account number. @param accountId @param module @throws FatalException	This method is used to update the commission recipient's name and Id it fetches the agent record based on the arrangementId and agentId , then updates the commission recipient's name and Id it is used when the commission Recipient of a account is changed
IAgentArrangementLinkageAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentArrangementLinkageAdapter	fetchFacilityForSubmissions	This method is used to fetch facility for a submission ID @param sessionContext @param submissionIds @return @throws FatalException	This method will return the facility linked to the Submissions account. @param sessionContext @param accountId @return @throws FatalException
ITaxationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationContentAdapter	printDepositCertificate		This method returns object of adapter class
ITaxationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationContentAdapter	printDepositCertificate		This method calls the OutboundDocumentApplicationService.generateSetOf Document for printing. @param sessionContext @param certificatePrintFactsDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchBasicPartyDetails		Returns the fetched PartyDTO. If PartyDTO doesn't exists returns null. @param partyId @return TrustDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchBasicPartyDetails		
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		Returns 'true' if PartyId is Valid for the Role Valuer, else returns 'false' @param valuerId @return PartyId Validity For Valuer.
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This method returns PartyDTO with AddressDTO for a partyID @param partyId @return partyDTO @throws FatalException
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This method gives a call to <b>CollateralValuationDTO</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralValuationDTO @return collateralValuationDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This method fetches party details for role. @param sessionContext @param partyId @param roleId @return fetchValuerDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		@param aAgentId @return
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		@param aValuerId @return
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This is the method for validating a valuer. @param valuerID @return
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This method gives a call to <b>&lt;collateralTitleSearchDTO&gt;</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralTitleSearchDTO @return collateralTitleSearchDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetails		This method fetches the party details for role. @param sessionContext @param partyId @param roleId @return fetchAgentDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	isPartyValid		Returns 'true' if PartyId is Valid else, returns 'false' @param PartyId @return isValid
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsWithAddress		This method returns PartyDTO with AddressDTO for a partyID @param partyId @return partyDTO @throws FatalException
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		This method gives a call to <b>CollateralValuationDTO</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralValuationDTO @return collateralValuationDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		This method fetches party details for role. @param sessionContext @param partyId @param roleId @return fetchValuerDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		@param aAgentId @return
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		@param aValuerId @return
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		This method gives a call to <b>&lt;b&gt;CollateralTitleSearchDTO&lt;/b&gt;</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralTitleSearchDTO @return collateralTitleSearchDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyDetailsForRole		This method fetches the party details for role. @param sessionContext @param partyId @param roleId @return fetchAgentDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method gives a call to <b>&lt;b&gt;CollateralValuationDTO&lt;/b&gt;</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralValuationDTO @return collateralValuationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method fetches party details for role. @param sessionContext @param partyId @param roleId @return fetchValuerDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method fetches Party Address. @param sessionContext @param partyID @return response
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		Fetch Party Address @param sessionContext @param partyAddressDTO @return partyAddressResponse @throws FatalException
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method gives a call to <b>CollateralTitleSearchDTO</b> to get the value of Party Name and Party Address. @param sessionContext @param collateralTitleSearchDTO @return collateralTitleSearchDTO
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method fetches the party details for role. @param sessionContext @param partyId @param roleId @return fetchAgentDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	fetchPartyAddress		This method fetches Party Address. @param sessionContext @param partyID @return response
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	getPartyClassName		This method return all party role as valuer
ICollateralValuationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationPartyAdapter	getPartyClassName		This method return all party id(s) who in role agent @return
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	fetchUserIdentificationDetails		This method is used to fetch the identificationDetails of a party. @param sessionContext @param partyId @return @throws FatalException
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	handlePartyMergeRequest		The method is used to handle merge in other related modules (e.g CASA,TD etc.) in party merge process. Any error or result gets added to the partyMergeResponse and the same is returned back. @param partyMergeResponse @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	handlePartyMergeRequest		This method is used for intimating other modules which are observer of Party Merge about party merge.
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateUserId		This method is called to merge following data of Parties : <ul> <li>Party Names <li>Party Demographic Details <li>Party Identification Details <li>Party Addresses <li>Party Contact Points <li>Party Roles <li>Party Warning Indicators <li>Party Comments <li>Party Photo <li>Party Signature <li>Party Credit Assessments <li>Party Bureau Reports <li>Update login Details </ul> In case any validation Error is found, it gets converted to InformationDTO and gets returned from the method.  @param sessionContext @param partyMergeDTO @throws FatalException
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateAgentArrangementLinkageForRoleWithNewPartyAndAgent		The method is used to update the AgentArrangementLinkage and ServiceProviderStatus of Victim's role with Survivors details. @param survivorPartyId @param victimPartyId @param roleId @param isRoleAssigned
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateAgentArrangementLinkageForRoleWithNewPartyAndAgent		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateAgentArrangementLinkageForRoleWithNewParty		The method is used to update the AgentArrangementLinkage and ServiceProviderStatus of Victim's role with Survivors details. @param survivorPartyId @param victimPartyId @param roleId @param isRoleAssigned
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateAgentArrangementLinkageForRoleWithNewParty		
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateAgentArrangementLinkageForRoleWithNewParty		
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateCommissionRecipientForAgentArrangementLinkage		The method is used to update the AgentArrangementLinkage and ServiceProviderStatus of Victim's role with Survivors details. @param survivorPartyId @param victimPartyId @param roleId @param isRoleAssigned
IPartyMergeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyMergeAdapter	updateCommissionRecipientForAgentArrangementLinkage		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationRuleApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationRuleApplicationAdapter	executeRule		This method is used to execute the rule which is created at time of rule maintenance. This method will internally call {@link ITaxationRuleApplicationAdapter} adapter to execute the rule by passing rule id, facts and process date as inputs. @param ruleID @param partyInformation @param processDate @return ArrayList<OutcomeDetails>
IChannelLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelLimitAdapter	validateLimit		This method is used fetch the adapter
IChannelLimitAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelLimitAdapter	validateLimit		This methods validates the account access, and if customer can do a transaction based on particular limit which has been set for that particular channel @param sessionContext the session context to be used @param channelContext the channel context to be used for validating @param accountID the account id which needs to be validated against @param response @param data the business context data to be used in validating @return @throws FatalException
IPartySecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartySecurityAdapter	fetchSecurityServicesResponse		
IPartySecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartySecurityAdapter	fetchSecurityServicesResponse		The following method is used to fetch UserAccessDTO. @param partySecurityAdapter @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartySecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartySecurityAdapter	fetchSecurityServicesResponse		
IPartySecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartySecurityAdapter	inquireUser		Returns all the roles in which the party exists in the system. @param party {@link com.ofss.fc.domain.party.entity.core.Party} @return array of {@link com.ofss.fc.app.party.dto.role.RoleDTO} @see {@link com.ofss.fc.domain.party.service.role.RoleService#fetchAllRolesAssignedToParty(IRole)}
IPartySecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartySecurityAdapter	inquireUser		
IOriginationSMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationSMSAdapter	inquireUserPartyId	This service inquires the party Id of the user @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @return The partyId of the user @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAccountRelationshipTypeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPartyAccountRelationshipTypeAdapter	fetchAllAccountsOfParty		
IPartyAccountRelationshipTypeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPartyAccountRelationshipTypeAdapter	fetchAllPartiesOfAccount		
IClearingBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\global\IClearingBatchAdapter	fetchClearingInstitutionBranches		
IClearingBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\global\IClearingBatchAdapter	fetchClearingInstitutionBranches		
IClearingBatchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\global\IClearingBatchAdapter	fetchClearingInstitutionBranches		
IJournalAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalAccountingAdapter	createEntries	@param sessionContext @param entries @param objectMap @return AccountingTransactionAdapterType @throws com.ofss.fc.infra.exception.FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalAccountingAdapter	createEntries	@param sessionContext @param entries @param objectMap @return AccountingTransactionAdapterType @throws com.ofss.fc.infra.exception.FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalAccountingAdapter	reverseEntries	@param sessionContext @param entries @param objectMap @return AccountingTransactionAdapterType @throws com.ofss.fc.infra.exception.FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalAccountingAdapter	reverseEntries	@param sessionContext @param entries @param objectMap @return AccountingTransactionAdapterType @throws com.ofss.fc.infra.exception.FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalAccountingAdapter	getSingleEntryDTOFromAccountingEntry	@param batchCode @param branchCode @return @throws FatalException	This method does the Batch Balancing By passing Suspense Entries for Un balanced Batches @param sessionContext @param journalEntryBatch @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill loan settlement details in case the chose settlement type is LOAN. @param priority @param splitValue @param loanAccount @param offerId @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill TD settlement details in case the chose settlement type is TDA. @param priority @param splitValue @param arrangementId @param depositID

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill own savings account settlement details in case the chose settlement type is DDAO. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill savings account settlement details in case the chose settlement type is DDA. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill Other savings account settlement details in case the chose settlement type is DDAT. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountCurrencyForAccount		This is a internal method will be called after initialize method. Basically It is being used to fill own savings account settlement details in case the chose settlement type is DDAO. @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyAccount		This is a internal method will be called after initialize method. Basically It is being used to fill external account settlement details in case the chose settlement type is PYMT . @param priority @param splitValue @param counterPartyIdentifier @param creditorName

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyAccount		This is a internal method will be called after initialize method. Basically It is being used to fill collection settlement details in case the chose settlement type is COLL . @param priority @param splitValue @param payeeAccount @param creditorName
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountType		This is a internal method will be called after initialize method. Basically It is being used to fill collection settlement details in case the chose settlement type is COLL . @param priority @param splitValue @param payeeAccount @param creditorName
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchAccountType		This is a internal method will be called after initialize method. Basically It is being used to fill collection settlement details in case the chose settlement type is CLRG . @param priority @param splitValue @param payeeAccount @param creditorName
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		<p>This is the method that will be called from transactions widget of dashboard. It will be used when SI setup request will be made for a account. SI request can be made for internal CASA as well as external account </p> @param sessionContext @param siRequestDTO @return @throws FatalException
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ICommonSettlementAdapter	fetchInstitutionDetailsForCounterPartyID		<p>This is the method that will be called from transactions widget of dashboard. It will be used when SI setup request will be made for an account. SI request can be made for internal CASA as well as external account </p> @param sessionContext @param siRequestDTO @return @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method will populate document List. @param sessionContext @param docList @return documentList @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	Validate document dto. Validates whether passed dto is not null that some value is passed for document creation and not contains any invalid document type. @param documentDTO the document dto @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	Fetches the documents for the given contentIds @param contentIdList @param documentDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	For each of the document categories,iterating through the categorizedDocumentType and fetching the respective documentType. The document type received is added as a part of categorizedDocumentTypeDTO @param sessionContext the session context @param documentTypeId the document type id @return the document type dto @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method will generate the data required for generation of the Document. Generation steps are as follows: <P> 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData <P> 2. Use the DocumakerDocumentAssembler to transform the extractData DTO into the IDocumakerModel.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	Deliver document for party. This method fetches the delivery details for party in case dispatched details not given by the caller. party address can be found from two ways 1.address details 2.contact point. Address details gives the postal address and contact point gives the parties email id or FAX number. Mandatory inputs require that-> Document DTO must be filled with bytes in OutboundDocumentContentDTO owner id. owner id must be already onboarded as party in flex-cube. @param sessionContext the session context @param documentDTO the document dto @return the document delivery response @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocumentType	This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext,, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is to retrieve the contentBytes from the contentId. @param sessionContext the session context @param outboundDocumentRequestDTO the outbound document request dto @param documentReferenceId the doc ref id @return the content bytes @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Gets the content bytes from document reference id. @param sessionContext the session context @param contentApplicationService the content application service @param documentReferenceId the document reference id @return the content bytes from document reference id @throws FatalException the fatal exception</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to generate the application form based on the product group serial number and the submissionInputDTO which includes the submission details such as submission ID, ApplicantDataDTO and ProductDetailsDTO.&lt;br&gt; &lt;br&gt; A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, ,submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the document is to be generated.In case it is null, the document will be generated for all the product group serial numbers of the given submissionInputDTO.</p> <p>@return {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} This class holds an array of the ApplicationFormDocumentDTO which contains the documentId of the newly created document with its submissionId and product group serial number. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} for more details.</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to generate the secondary auxiliary indexes for an already generated document based on the product group serial number and the submissionInputDTO which includes the ProductDetailsDTO from which we obtain the document details for the various products.&lt;br&gt; &lt;br&gt; A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the indexes are to be generated. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method updates the AuxillaryDocumentIndexDTO for all the documents within a submission for which the secondary auxiliary indexes are generated by fetching such documents through their document Id's. @param sessionContext @param indexDTOMap @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will populate document List. @param sessionContext @param docList @return documentList @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will populate documaker request in case of Redispatch. OBPSUPPFIX23-603 Modified the method to merge the changes to release R2.3 @param sessContext @param domainObjectDTO @return DoPublishFromFactoryRequest @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method updates the document details. &lt;li&gt;Firstly the Document is fetched using the Document Id.&lt;/li&gt;&lt;li&gt; Before Updating Content it made checks for the Document Status Type and Execution Status Type,The Combination Of these values is checked&lt;/li&gt;&lt;li&gt;After validating that the Document Content call is given to update Document of the Document Service.&lt;/li&gt;</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,The documentDTO which would be persisted to DB</p> <p>@return {@link com.ofss.fc.app.content.dto.DocumentContentResponse}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to update the image in IPM,New image added against the same content id with the new version. Document data is also updated against document id in DMS. Mandatory fields are: &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@linkplain #documentId}: Mandatory</li> <li>&lt;li&gt;{@linkplain #documentReferenceId}: Mandatory</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; Here we can provide the functionality of fetching content id using document id but we may not know that doc id have the content id or not.hence caller need to pass both values. @param context @param metaDataValue @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method sets the contentId of a document in the DB. In cases where incorrect content is attached to a document,this method can be used. First the document is fetched using the documentId and then the documentReferenceId is set to null. Then DocumentService.updateDocument() is called.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,The documentId for which the contentId is to be set to null @return {@link com.ofss.fc.service.response.TransactionStatus}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Gets the val. @param sessionContext the session context @param documentDTO the document dto @param val the val @param documentApplicationService the document application service @return the val @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Validate document dto.Validates whether passed dto is not null that some value is passed for document creation and not contains any invalid document type. @param documentDTO the document dto @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	Fetches the documents for the given contentIds @param contentIdList @param documentDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method checks whether the contentId is referred by any other document.If it is, then the content cannot be deleted @param document @param sessionContext @param documentService @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method merges the party id in the document audit details. @param partyIdToMerge @param partyId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method merges the party id in the document and auxiliary document details. @param partyIdToMerge @param partyId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>For each of the document categories,iterating through the categorizedDocumentType and fetching the respective documentType. The document type received is added as a part of categorizedDocumentTypeDTO @param sessionContext the session context @param documentTypeId the document type id @return the document type dto @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext,, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will generate the data required for generation of the Document. Generation steps are as follows: &lt;P&gt; 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData &lt;P&gt; 2. Use the DocumakerDocumentAssembler to transform the extractData DTO into the IDocumakerModel.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Deliver document for party. This method fetches the delivery details for party in case dispatched details not given by the caller. party address can be found from two ways 1.address details 2.contact point. Address details gives the postal address and contact point gives the parties email id or FAX number. Mandatory inputs require that-&gt; Document DTO must be filled with bytes in OutboundDocumentContentDTO owner id. owner id must be already onboarded as party in flex-cube. @param sessionContext the session context @param documentDTO the document dto @return the document delivery response @throws FatalException the fatal exception</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,documentDTO TODO: Explain the documentDTO above @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is to retrieve the contentBytes from the contentId. @param sessionContext the session context @param outboundDocumentRequestDTO the outbound document request dto @param documentReferenceId the doc ref id @return the content bytes @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	Gets the content bytes from document reference id. @param sessionContext the session context @param contentApplicationService the content application service @param documentReferenceId the document reference id @return the content bytes from document reference id @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to generate the application form based on the product group serial number and the submissionInputDTO which includes the submission details such as submission ID, ApplicantDataDTO and ProductDetailsDTO.&lt;br&gt; &lt;br&gt; A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, ,submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the document is to be generated.In case it is null, the document will be generated for all the product group serial numbers of the given submissionInputDTO.</p> <p>@return {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} This class holds an array of the ApplicationFormDocumentDTO which contains the documentId of the newly created document with its submissionId and product group serial number. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} for more details.</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to generate the secondary auxiliary indexes for an already generated document based on the product group serial number and the submissionInputDTO which includes the ProductDetailsDTO from which we obtain the document details for the various products.&lt;br&gt; &lt;br&gt; A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document. &lt;br&gt; &lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, ,This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, ,Array contains all the product group serial numbers for which the indexes are to be generated. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method updates the AuxillaryDocumentIndexDTO for all the documents within a submission for which the secondary auxiliary indexes are generated by fetching such documents through their document Id's. @param sessionContext @param indexDTOMap @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This method will populate document List. @param sessionContext @param docList @return documentList @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will populate documaker request in case of Redispatch. OBPSUPPFIX23-603 Modified the method to merge the changes to release R2.3 @param sessContext @param domainObjectDTO @return DoPublishFromFactoryRequest @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method updates the document details. &lt;li&gt;Firstly the Document is fetched using the Document Id.&lt;/li&gt;&lt;li&gt; Before Updating Content it made checks for the Document Status Type and Execution Status Type,The Combination Of these values is checked&lt;/li&gt;&lt;li&gt;After validating that the Document Content call is given to update Document of the Document Service.&lt;/li&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,The documentDTO which would be persisted to DB @return {@link com.ofss.fc.app.content.dto.DocumentContentResponse}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used to update the image in IPM,New image added against the same content id with the new version. Document data is also updated against document id in DMS. Mandatory fields are: &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@linkplain #documentId}: Mandatory</li> <li>&lt;li&gt;{@linkplain #documentReferenceId}: Mandatory</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; Here we can provide the functionality of fetching content id using document id but we may not know that doc id have the content id or not.hence caller need to pass both values. @param context @param metaDataValue @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method sets the contentId of a document in the DB. In cases where incorrect content is attached to a document,this method can be used. First the document is fetched using the documentId and then the documentReferenceId is set to null. Then DocumentService.updateDocument() is called.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,The documentId for which the contentId is to be set to null @return {@link com.ofss.fc.service.response.TransactionStatus}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Gets the val. @param sessionContext the session context @param documentDTO the document dto @param val the val @param documentApplicationService the document application service @return the val @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	Validate document dto.Validates whether passed dto is not null that some value is passed for document creation and not contains any invalid document type. @param documentDTO the document dto @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId, ,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Fetches the documents for the given contentIds @param contentIdList @param documentDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method checks whether the contentId is referred by any other document.If it is, then the content cannot be deleted @param document @param sessionContext @param documentService @return @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method merges the party id in the document audit details. @param partyIdToMerge @param partyId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method merges the party id in the document and auxiliary document details. @param partyIdToMerge @param partyId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>For each of the document categories,iterating through the categorizedDocumentType and fetching the respective documentType. The document type received is added as a part of categorizedDocumentTypeDTO @param sessionContext the session context @param documentTypeId the document type id @return the document type dto @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method will generate the data required for generation of the Document. Generation steps are as follows: &lt;P&gt; 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData &lt;P&gt; 2. Use the DocumakerDocumentAssembler to transform the extractData DTO into the IDocumakerModel.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the fetchDocumentType method of the entity. @return {@link com.ofss.fc.app.content.dto.DocumentTypeInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method gives the service return instance of BIP services if the document template type is BIP(B) and return Documaker services if the document template Type is documaker(D). For this method we need to pass SessionContext and {@link DocumentTypeDTO} @param sessionContext @param documentTypeDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>Deliver document for party. This method fetches the delivery details for party in case dispatched details not given by the caller. party address can be found from two ways 1.address details 2.contact point. Address details gives the postal address and contact point gives the parties email id or FAX number. Mandatory inputs require that-&gt; Document DTO must be filled with bytes in OutboundDocumentContentDTO owner id. owner id must be already onboarded as party in flex-cube. @param sessionContext the session context @param documentDTO the document dto @return the document delivery response @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchDocument	<p>This represents the fetchDocument method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,documentId,documentId The unique id by which the document is identified and indexed in OBP @return {@link com.ofss.fc.app.content.capture.dto.DocumentInquiryResponse}</p> <p>@throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	This method is used to generate the application form based on the product group serial number and the submissionInputDTO which includes the submission details such as submission ID, ApplicantDataDTO and ProductDetailsDTO.    A request is sent to the DMS module by populating the OutboundDocumentRequestDTO which generates the document.     @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,SubmissionInputDTO, submissionInputDTO, This class consists of details of submission including the ApplicantDataDTO, ProductDetailsDTO and the HeaderDTO containing information such as application branch, data entry branch, submissionId, creation date, etc. @fcb.param Integer[],productGroupserialNos, Array contains all the product group serial numbers for which the document is to be generated.In case it is null, the document will be generated for all the product group serial numbers of the given submissionInputDTO. @return {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} This class holds an array of the ApplicationFormDocumentDTO which contains the documentId of the newly created document with its submissionId and product group serial number. Refer to {@link com.ofss.fc.app.origination.dto.core.submission.ApplicationFormGenerationResponse} for more details. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	Generate document. @param sessionContext the session context @param documentInputDTO the document input DTO @return the com.ofss.fc.service.response.transaction status @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	This method will Generate set of document with parameter sessionContext and outboundDocumentRequestDTOSet. Generation steps are as follows: <P> 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData. <P> 2. Use the DocumakerDocumentHandler for transform the extractData DTO into the iExtract. @param sessionContext the session context @param outboundDocumentRequestDTOSet the out bound document request dto set @return the com.ofss.fc.service.response.transaction status @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	This method will generate the requested Documents and will save the complete list of documents.From UI user selects multiple document to be generated. This methods takes input as opoutboundDocumentListRecordDTo and converts into the OutboundDocumentRequestDTO.if multiple records selected then then for each ListRecordDTO OutboundDocumentRequestDTO created.In OutboundDocumentRequestDTO there is array of DocumentPolicyDetailDTO but it should e of size one.i.e for generation only one facts require hence policy detail should be consider as one. @param sessionContext the session context @param outboundDocumentListDTO the out bound document list dto @return the document generation response set @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	generateSetOfDocument	This service is called to generate the set of documents @param sessionContext @param outboundDocumentRequestDTOSet @return @throws FatalException	This method will Generate set of document with parameter sessionContext and outboundDocumentRequestDTOSet. Generation steps are as follows: <P> 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData. <P> 2. Use the DocumakerDocumentHandler for transform the extractData DTO into the iExtract. @param sessionContext the session context @param outboundDocumentRequestDTOSet the out bound document request dto set @return the com.ofss.fc.service.response. transaction status @throws FatalException the fatal exception



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateBatchSetOfDocument	This service is called to generate the set of documents for batch @param sessionContext @param fileName @return @throws FatalException	
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateBatchSetOfDocument	This service is called to generate the set of documents for batch @param sessionContext @param fileName @return @throws FatalException	This service is called to generate the set of documents for batch @param sessionContext @param fileName @return @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	generateBatchSetOfDocument	This service is called to generate the set of documents for batch @param sessionContext @param fileName @return @throws FatalException	Method generateBatchReport is a wrapper method which Calls generateReport method of FCRJReporter class to generate report @param p_con Connection object @param p_xml_bp FCRJXMLParser @param p_xpath String
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchRequestId		This method will make entry of created document in DMS tables.(FLX_CM_DOC) as well as Audit Table. @param sessionContext the session context @param OutboundDocumentRequestDTO the out bound document request dto set @return documentGenerationResponse @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchRequestId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchRequestId		This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchRequestId		In this method the input in bytes of the Document. It Converts the document content bytes into file and delivers it to destinations. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchRequestId		
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	archiveDocument		This method will make entry of created document in DMS tables.(FLX_CM_DOC) as well as Audit Table. @param sessionContext the session context @param OutboundDocumentRequestDTO the out bound document request dto set @return documentGenerationResponse @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	archiveDocument		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fillAuditData		This method will make entry of created document in DMS tables.(FLX_CM_DOC) as well as Audit Table. @param sessionContext the session context @param OutboundDocumentRequestDTO the out bound document request dto set @return documentGenerationResponse @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fillAuditData		
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fillAuditData		This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fillAuditData		In this method the input in bytes of the Document. It Converts the document content bytes into file and delivers it to destinations. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fillAuditData		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	audit		1.IPP_HOST=required field 2.IPP_PRINTER_NAME=required field @param outboundDocumentRequestDTO the outbound document request dto @param contentBytes the content bytes @return the document delivery response @throws DeliveryException the delivery exception @throws IOException Signals that an I/O exception has occurred. @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	audit		This method will make entry of created document in DMS tables.(FLX_CM_DOC) as well as Audit Table. @param sessionContext the session context @param OutboundDocumentRequestDTO the out bound document request dto set @return documentGenerationResponse @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	audit		
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	audit		This method is used for delivering the Document which is already present in the IPM. We just need to fetch the Document from the IPM and send it to the required Destination. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	audit		In this method the input in bytes of the Document. It Converts the document content bytes into file and delivers it to destinations. @param sessionContext the session context @param outboundDocumentRequestDTO the out bound document request dto @return the document delivery response @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	audit		
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getNewDocumentId		This method will Generate set of document with parameter sessionContext and outboundDocumentRequestDTOSet. Generation steps are as follows: <P> 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData. <P> 2. Use the DocumakerDocumentHandler for transform the extractData DTO into the iExtract. @param sessionContext the session context @param outboundDocumentRequestDTOSet the out bound document request dto set @return the com.ofss.fc.service.response. transaction status @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getNewDocumentId		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getApplicationIdForInsert	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getApplicationIdForInsert	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	Adds a new document to the repository. Returns back the document Id as generated in the repository. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, ,This object contains the meta data values which are used while persisting the document in IPM. @fcb.param MI,String,contentData, ,This is the content that is uploaded in IPM. @return {@link String} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	Adds a new document to the repository. Returns back the document Id as generated in the repository. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, ,This object contains the meta data values which are used while persisting the document in IPM. @fcb.param MI,String,contentData, ,This is the content that is uploaded in IPM. @return {@link String} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	Searches for definition of property and meta data. This service can be called to load the search page. Returns back a collection of Definition. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentTypeDTO,documentType, ,This object contains the document type which is used for searching the definition.Using the documentType, the applicationId is fetched which is then used for searching the meta data definition. @return {@link com.ofss.fc.ixface.content.dto.ContentResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	This method will Generate set of document with parameter sessionContext and outboundDocumentRequestDTOSet. Generation steps are as follows: <P> 1. Based on the Document Type and the Module Name, from the outboundDocumentRequestDTOSet, invoke appropriate DataExtractor.extractData. <P> 2. Use the DocumakerDocumentHandler for transform the extractData DTO into the iExtract. @param sessionContext the session context @param outboundDocumentRequestDTOSet the out bound document request dto set @return the com.ofss.fc.service.response.transaction status @throws FatalException the fatal exception



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException
IIIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationContentAdapter	getApplicationId	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException	This method fetches the applicationId corresponding to the documentType. The mapping is stored available in DocumentTypeMetaData. @param documentTypeId @return @throws FatalException
IIIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationContentAdapter	fetchByMetadataId		This method searches for a document using a set of search parameters. <li>The searchData contains metadata values which are used to search a document in content server (e.g IPM)</li> <li>the documentDTO is used to search for a document in DB.</li> <li>Note that the documentDTO passed as input must contain the basic search parameters.</li>    Only the following fields would be used from the documentDTO -: <ul> <li>{@linkplain #indexType}: Optional <li>{@linkplain #indexValue}: Optional <li>{@linkplain #serviceEventId}: Optional <li>{@linkplain #ownerId}: Optional <li>{@linkplain #documentStatusType}: Optional </ul> </li> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentDTO,documentDTO, ,The documentDTO which would be persisted to DB

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@fcb.param MI,SearchData,searchData, ,This object contains the meta data values which are used while uploading the document in IPM. @fcb.param MI,String,contentData, ,This is the content that is uploaded in IPM. @return {@link com.ofss.fc.ixface.content.dto.SearchResultSet} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	fetchByMetaDataId		
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		This method is to retrieve the contentBytes from the contentId. @param sessionContext the session context @param outboundDocumentRequestDTO the outbound document request dto @param documentReferenceId the doc ref id @return the content bytes @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Gets the content bytes from document reference id. @param sessionContext the session context @param contentApplicationService the content application service @param documentReferenceId the document reference id @return the content bytes from document reference id @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		<p>Adds a new document to the repository. Returns back the document Id as generated in the repository.</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,This object contains the meta data values which are used while persisting the document in IPM. @fcb.param MI,String,contentData, ,This is the content that is uploaded in IPM. @return {@link String} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		<p>This method deletes content from IPM @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,The document instance. It contains the content id which is to be deleted.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		<p>This method updates the document details. &lt;li&gt;Firstly the Document is fetched using the Document Id.&lt;/li&gt;</p> <p>&lt;li&gt; Before Updating Content it made checks for the Document Status Type and Execution Status Type,The Combination Of these values is checked&lt;/li&gt;</p> <p>&lt;li&gt;After validating that the Document Content call is given to update Document of the Document Service.&lt;/li&gt;</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,DocumentDTO,documentDTO, ,The documentDTO which would be persisted to DB</p> <p>@return {@link com.ofss.fc.app.content.dto.DocumentContentResponse}</p>
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		<p>This method is used to update the image in IPM,New image added against the same content id with the new version. Document data is also updated against document id in DMS. Mandatory fields are: &lt;ul&gt;</p> <p>&lt;li&gt;{@linkplain #documentId}: Mandatory</p> <p>&lt;li&gt;{@linkplain #documentReferenceId}: Mandatory</p> <p>&lt;/ul&gt;</p> <p>&lt;br&gt; Here we can provide the functionality of fetching content id using document id but we may not know that doc id have the content id or not.hence caller need to pass both values. @param context @param metaDataValue @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		<p>This method will delete the content id(uploaded image) from content server (e.g IPM) and delete corresponding document from DB. &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@linkplain #documentId}: Mandatory</li> <li>&lt;li&gt;{@linkplain #documentReferenceId}: Mandatory</li> </ul> <p>&lt;/ul&gt; @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param MI,DocumentDTO,documentDTO, ,The document to be deleted @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		<p>Document upload.This method is third party method which is used to upload the document in DMS System. Their are various scenarios in which document can be uploaded in the system. In all the following scenarios bytes are always accepted in document dto. &lt;li&gt;If input (only bytes and supporting info one of auxiliary index or id given which is not exist in system) -&gt; create document in IPM and create data in DMS.&lt;/li&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If input (Document id)-&gt;System fetches first the document using document id and check document already exist or not. If document content id exist then update version of the document in IPM with new bytes. else create new document in IPM and update the document id for content id in DMS&lt;/li&gt;.</li> <li>&lt;li&gt;If input (bytes and indexes)-&gt;search document on the basis of indexes and get all the result. if (search result = 1) get document id and go to 2. else fatal exception.&lt;/li&gt;</li> </ul> <p>@param sessionContext the session context @param documentDTO the document dto</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Update doc in dms. @param sessionContext the session context @param documentDTO the document dto @param contentApplicationService the content application service @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Update document in ipm and dms.data from previous content id is fetched and assign to new one. @param sessionContext the session context @param documentDTO the document dto @param contentApplicationService the content application service @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Creates the new document. @param sessionContext the session context @param documentDTO the document dto @param documentRefId the document ref id @param contentApplicationService the content application service @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Gets the val. @param sessionContext the session context @param documentDTO the document dto @param val the val @param documentApplicationService the document application service @return the val @throws FatalException the fatal exception
IntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationContentAdapter	getDocumentReferenceId		Update content. @param context the context @param documentDTO the document dto @param transactionStatus the transaction status @return the string @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		Searches for a document in the DB DocumentDTO passed as input must contain the basic search parameters. Only the following fields would be used from the documentDTO IndexType,IndexValue,ServiceEventId,OwnerId and DocumentStatusType. This Type of Search is referred as the Basic Search. @param documentDTO @param searchResultSet @param searchData @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		This method merges the party id in the document and auxiliary document details. @param partyIdToMerge @param partyId @throws FatalException
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		This method is for delivering the Documents which are already saved in IPM. @param sessionContext the session context @param outboundDocumentRequestDTO the outbound document request dto @return the document delivery response @throws FatalException the fatal exception
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		@see com.ofss.fc.app.content.outbound.service.IDocumentDeliveryApplicationService#deliverGroupedDocuments (com.ofss.fc.app.context.SessionContext, java.util.List)
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		Fill inserts for mailing. @param sessionContext the session context @param documentDTOArray the document dto array @param listOfFilesToBeDelivered the list of files to be delivered @param inserts the inserts @throws FatalException the fatal exception @throws IOException Signals that an I/O exception has occurred.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	getDocumentReferenceId		
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchInsertsForDocument		@see com.ofss.fc.app.content.outbound.service.IDocumentDeliveryApplicationService#deliverGroupedDocuments (com.ofss.fc.app.context.SessionContext, java.util.List)
IIntegrationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationContentAdapter	fetchInsertsForDocument		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		Method will validate user define status initiated by User. User-Defined status can only be changed if the reason code is not due to KYC Unconfirmed,if the reason code is due to KYC Unconfirmed the following is the behaviour: <ul style="list-style-type: none"> <li>&lt;b&gt;If the new user defined status does not contain KYC Unconfirmed but the old user defined status contains KYC UnConfirmed status it means user cannot change the user defined status of KYC Unconfirmed through screen and this error is added to validation error,there can also be a case where the joint applicants KYC is Unconfirmed and hence the status cannot be changed.&lt;/b&gt;</li> <li>&lt;b&gt;If the new user defined status contains KYC Unconfirmed but the old user defined status does not contains KYC Unconfirmed it means user is trying to add status of KYC Unconfirmed which is not allowed through this service and hence the user defined status of KYC Unconfirmed through screen and this error is added to validation error.&lt;/b&gt;</li> <li>&lt;b&gt;If the new user defined status contains KYC Unconfirmed and the old user defined status contains KYC Unconfirmed it means there is a KYC Unconfirmed status this error is added to validation error.&lt;/b&gt;</li> </ul>
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		Demand deposit account has following types of status <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;System defined status: Always has one {@link AgingStatus}.System maintains this status.OE/teller is allowed to change this status only from {@link AgingStatus#Dormant} to {@link AgingStatus#Regular}.&lt;/li&gt; <li>User defined status: Can</li> </li></ul> </li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>have zero more {@link DemandDepositLifeCycleStatus}.OE/teller can add/delete status. &lt;/ol&gt; OE or teller can modify above account status using screen.This method is used to modify user define status of demand deposit account. This method will be used to remove {@link UserDefinedStatus}.&lt;br&gt;This method will perform following activities. &lt;ol&gt; &lt;li&gt;Fetch existing user defined status- &lt;ol&gt; &lt;li&gt;remove given UserDefineStatus from existing set of UserDefineStatus. &lt;li&gt;Call {@link AccountStatusManagementService#modifyAccountStatus()} with following parameter. &lt;ol&gt; &lt;li&gt;input.accountId &lt;li&gt;null &lt;li&gt;status created in step1 &lt;li&gt;&gt;false &lt;li&gt;{@link AccountStatusChangeType#UserDefinedStatus}. &lt;/ol&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,String,accountId, &lt;br&gt; The attribute required to modify accountStatus of DDAccount. @param DemandDepositLifecycleStatus user defined status to be removed for the account .&lt;br&gt; In case of failure - &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ul&gt; @throws FatalException @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		Refer BaseApplicationService.checkForJointApplicantPending @param applicationId @return @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		This method sets the account status based on the KYC status of the party  Setting the account status get the party KYC status. If it is confirmed then mark the account status as confirm else mark the account as AccountStatus.KYCUncconfirmed  If the KYC status of any of the holders is 'KYC Unconfirmed', open the TD account with status as 'Debit Block' with reason as 'KYC Unconfirmed'. @param applicationContext @param partyToAccountRelationshipDTOs
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	checkForJointApplicantPending		
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	constructSubmissionRequestForGivenFacilityId	This method constructs a {@link SubmissionInputDTO} for the given Facility Id. The facility id should be the id of the parent facility. @param sessionContext the {@link SessionException} @param facilityId the facility Id of the main parent facility @param accountCode the Account Id of the account opened under the facility @return {@link SubmissionInputDTO} @throws FatalException	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	constructSubmissionRequestForGivenFacilityId	This method constructs a {@link SubmissionInputDTO} for the given Facility Id. The facility id should be the id of the parent facility. @param sessionContext the {@link SessionException} @param facilityId the facility Id of the main parent facility @param accountCode the Account Id of the account opened under the facility @return {@link SubmissionInputDTO} @throws FatalException	This method creates a new Account Variation Request for the given account code.  It first fetches the parent facility linked to the given account.  Using this facility, it creates a Submission Request. This submission request is linked to the account variation request and is persisted into the database. It returns the newly created account variation request and the serialized DTO @param sessionContext the {@link SessionContext} @param accountCode the Account Code @return the newly created account variation request and the serialized DTO @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	constructSubmissionRequestForGivenFacilityId	This method constructs a {@link SubmissionInputDTO} for the given Facility Id. The facility id should be the id of the parent facility. @param sessionContext the {@link SessionException} @param facilityId the facility Id of the main parent facility @param accountCode the Account Id of the account opened under the facility @return {@link SubmissionInputDTO} @throws FatalException	@see IAccountOriginationAdapter#constructSubmissionRequestForGivenFacilityId(SessionContext, String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	saveSubmissionAsDraft	<p>This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details. &lt;br&gt; The submission draft can be retrieved in the following ways: &lt;br&gt; 1. My Tracker - This tracker is used by the applicant to log into the system. &lt;br&gt; 2. Application Tracker - This tracker is used by the banking associated to log into the system. &lt;br&gt; For systems where all applicants applying for the submission are new, at least one primary applicant must be specified for the system. When the submission is saved, the primary applicant is on-boarded as a 'Member' in the party system. The process takes care of on-boarding process&lt;br&gt; The primary applicant can later log into the system and submit the application or the banker can retrieve the submission using this primary applicant and continue with the submission. The inputs submissionInputDTO will be serialized into String and passed to DraftApplicationService for save as</p>	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>draft. &lt;br&gt; @fcb.param  MI,SessionContext,sessionContext,  ,Session Context which must contain  the session context information  such as user id, branch, branch code,  channel etc. @fcb.param  MI,SubmissionProcessDraftDTO,  submissionProcessDraftDTO, ,This  class holds the entire information of  one submission and contains the  input details as well as the  processed output from the process  @return {@link  com.ofss.fc.service.response.Transac  tionStatus} An instance of the  transaction status which contains  the required information about the  status of execution indicating if the  invocation succeeded or failed.  @exception  OriginationErrorConstants.RULE_I  D_NOT_MAINTAINED :  FC_OR_00011 &lt;br&gt; This exception  is thrown when Rule Id is not linked  in Origination Preference&lt;br&gt; for  LRM Submission Type and LRM  Provider Rules.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	saveSubmissionAsDraft	<p>This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details. &lt;br&gt; The submission draft can be retrieved in the following ways: &lt;br&gt; 1. My Tracker - This tracker is used by the applicant to log into the system. &lt;br&gt; 2. Application Tracker - This tracker is used by the banking associated to log into the system. &lt;br&gt; For systems where all applicants applying for the submission are new, at least one primary applicant must be specified for the system. When the submission is saved, the primary applicant is on-boarded as a 'Member' in the party system. The process takes care of on-boarding process&lt;br&gt; The primary applicant can later log into the system and submit the application or the banker can retrieve the submission using this primary applicant and continue with the submission. The inputs submissionInputDTO will be serialized into String and passed to DraftApplicationService for save as</p>	<p>This method updates an already existing account variation request and the {@link VariationRequestDTO} in a serialized form @param sessionContext the {@link SessionContext} @param accountVariationRequestDTO the {@link AccountVariationRequestDTO} @param variationRequestDTO the {@link VariationRequestDTO} @return the {@link TransactionStatus} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>draft. &lt;br&gt; @fcb.param  MI,SessionContext,sessionContext,  ,Session Context which must contain  the session context information  such as user id, branch, branch code,  channel etc. @fcb.param  MI,SubmissionProcessDraftDTO,  submissionProcessDraftDTO, ,This  class holds the entire information of  one submission and contains the  input details as well as the  processed output from the process  @return {@link  com.ofss.fc.service.response.Transac  tionStatus} An instance of the  transaction status which contains  the required information about the  status of execution indicating if the  invocation succeeded or failed.  @exception  OriginationErrorConstants.RULE_I  D_NOT_MAINTAINED :  FC_OR_00011 &lt;br&gt; This exception  is thrown when Rule Id is not linked  in Origination Preference&lt;br&gt; for  LRM Submission Type and LRM  Provider Rules.</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	saveSubmissionAsDraft	<p>This service is used for saving submission details in draft. A user may select to save and exit and return to the submission later. In this case, the details that were entered previously can be retrieved and the user can continue with the flow. Details for this draft can be retrieved using the submission identifier used for saving the details. &lt;br&gt; The submission draft can be retrieved in the following ways: &lt;br&gt; 1. My Tracker - This tracker is used by the applicant to log into the system. &lt;br&gt; 2. Application Tracker - This tracker is used by the banking associated to log into the system. &lt;br&gt; For systems where all applicants applying for the submission are new, at least one primary applicant must be specified for the system. When the submission is saved, the primary applicant is on-boarded as a 'Member' in the party system. The process takes care of on-boarding process&lt;br&gt; The primary applicant can later log into the system and submit the application or the banker can retrieve the submission using this primary applicant and continue with the submission. The inputs submissionInputDTO will be serialized into String and passed to DraftApplicationService for save as</p>	<p>@see IAccountOriginationAdapter#saveSubmissionAsDraft(SessionContext, SubmissionProcessDraftDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>draft. &lt;br&gt; @fcb.param  MI,SessionContext,sessionContext,  ,Session Context which must contain  the session context information  such as user id, branch, branch code,  channel etc. @fcb.param  MI,SubmissionProcessDraftDTO,  submissionProcessDraftDTO, ,This  class holds the entire information of  one submission and contains the  input details as well as the  processed output from the process  @return {@link  com.ofss.fc.service.response.Transac  tionStatus} An instance of the  transaction status which contains  the required information about the  status of execution indicating if the  invocation succeeded or failed.  @exception  OriginationErrorConstants.RULE_I  D_NOT_MAINTAINED :  FC_OR_00011 &lt;br&gt; This exception  is thrown when Rule Id is not linked  in Origination Preference&lt;br&gt; for  LRM Submission Type and LRM  Provider Rules.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	evaluateLRMSubmissionType	This method will be invoked to evaluate LRMSubmissionType rule to get the Submission type for the LRM policy. This will be called in the pre-assessment stage. @param sessionContext @param submissionInputDTO @param productGroupSerialNo @return @throws FatalException	This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	evaluateLRMSubmissionType	This method will be invoked to evaluate LRMSubmissionType rule to get the Submission type for the LRM policy. This will be called in the pre-assessment stage. @param sessionContext @param submissionInputDTO @param productGroupSerialNo @return @throws FatalException	This method is used to fetch the LRMSubmissionType for the LRM arrangement @param sessionContext the {@link SessionContext} @param submissionInputDTO the {@link SubmissionInputDTO} @return {@link LRMRuleExecutionResponse} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	evaluateLRMSubmissionType	This method will be invoked to evaluate LRMSubmissionType rule to get the Submission type for the LRM policy. This will be called in the pre-assessment stage. @param sessionContext @param submissionInputDTO @param productGroupSerialNo @return @throws FatalException	@see IAccountOriginationAdapter#evaluateLRMSubmissionType(SessionContext, SubmissionInputDTO, Integer)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	performConditionComplianceCheck	This method checks if the preliminary conditions for the borrowing entity have been complied @param sessionContext the {@link SessionException} @param borrowingEntityId the Borrowing Entity Id @param applicationId the Application Id @param prodGroupLinkage the {@link LendingProductGroupLinkageDTO} @return the {@link ConditionsComplianceCheckResponse} @throws FatalException	This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	performConditionComplianceCheck	This method checks if the preliminary conditions for the borrowing entity have been complied @param sessionContext the {@link SessionException} @param borrowingEntityId the Borrowing Entity Id @param applicationId the Application Id @param prodGroupLinkage the {@link LendingProductGroupLinkageDTO} @return the {@link ConditionsComplianceCheckResponse} @throws FatalException	This method checks if the preliminary conditions for the borrowing entity have been complied @param sessionContext the {@link SessionException} @param borrowingEntityId the Borrowing Entity Id @param applicationId the Application Id @param prodGroupLinkage the {@link LendingProductGroupLinkageDTO} @return the {@link ConditionsComplianceCheckResponse} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	performConditionComplianceCheck	This method checks if the preliminary conditions for the borrowing entity have been complied @param sessionContext the {@link SessionException} @param borrowingEntityId the Borrowing Entity Id @param applicationId the Application Id @param prodGroupLinkage the {@link LendingProductGroupLinkageDTO} @return the {@link ConditionsComplianceCheckResponse} @throws FatalException	@see IAccountOriginationAdapter#performConditionComplianceCheck(SessionContext, String, String, LendingProductGroupLinkageDTO)
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	withdrawAccountVariation	This method will withdraw the account variation. @param withdrawSubmissionDetailsDTO @param submissionId @param isSubmissionFeeNegotiated @return @throws FatalException	This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	withdrawAccountVariation	This method will withdraw the account variation. @param withdrawSubmissionDetailsDTO @param submissionId @param isSubmissionFeeNegotiated @return @throws FatalException	This method withdraws an account variation request which is in progress @param sessionContext the {@link SessionException} @param variationRequestId the Variation Request Id @param withdrawSubmissionDetailsDTO the {@link WithdrawSubmissionDetailsDTO} @return the {@link TransactionStatus} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	withdrawAccountVariation	This method will withdraw the account variation. @param withdrawSubmissionDetailsDTO @param submissionId @param isSubmissionFeeNegotiated @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	abortInstancesforAccountVariation	This method will abort the account variation process for the input submission Id. @param submissionId @throws FatalException	This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	abortInstancesforAccountVariation	This method will abort the account variation process for the input submission Id. @param submissionId @throws FatalException	This method will abort an Account Variation Request @param sessionContext the {@link SessionException} @param variationRequestId the Variation Request Id @return the {@link TransactionStatus} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	abortInstancesforAccountVariation	This method will abort the account variation process for the input submission Id. @param submissionId @throws FatalException	
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	markVariationSubmissionAsCompleted	This method will mark the submission and application for the account variation request as completed and processed respectively. @param submissionId the Submission / Variation Request Id @throws FatalException	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	markVariationSubmissionAsCompleted	This method will mark the submission and application for the account variation request as completed and processed respectively. @param submissionId the Submission / Variation Request Id @throws FatalException	This method will confirm an Account Variation Request.  It will first confirm the proposed facilities for the variation request  It will then do the variation request transaction depending the variation request type. For eg., for {@link VariationRequestType#LOAN_ROLLOVER_REQUEST}, it will call the Loan module's {@link IAccountLoanAdapter#addLoanRolloverInstruction(SessionContext, com.ofss.fc.app.loan.dto.rolloverinstruction.LoanRolloverInstructionDTO, com.ofss.fc.app.loan.dto.rate.lock.RateLockDTO, com.ofss.fc.app.loan.dto.fee.collection.FeeCollectionDTO)} .  It will then mark the variation request status as completed. @param sessionContext the {@link SessionException} @param variationRequestId the Variation Request Id @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	markVariationSubmissionAsCompleted	This method will mark the submission and application for the account variation request as completed and processed respectively. @param submissionId the Submission / Variation Request Id @throws FatalException	This method will mark the submission and application for the account variation request as completed and processed respectively. @param submissionId the Submission / Variation Request Id @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	updateVariationSubmissionStatus	This method will update the submission status of the account variation. @param submissionId @param submissionStatusType	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	updateVariationSubmissionStatus	This method will update the submission status of the account variation. @param submissionId @param submissionStatusType	This method will update the submission status of the account variation. @param sessionContext the {@link SessionException} @param variationRequestId the Variation Request Id @return the {@link TransactionStatus} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	updateVariationSubmissionStatus	This method will update the submission status of the account variation. @param submissionId @param submissionStatusType	
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	validateSubmission	This service will validate all inputs within the submission. @param sessionContext the {@link SessionException} @param submissionInputDTO the {@link SubmissionInputDTO} @return the {@link TransactionStatus} @throws FatalException	This method returns the Adapter for calling the origination module
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	validateSubmission	This service will validate all inputs within the submission. @param sessionContext the {@link SessionException} @param submissionInputDTO the {@link SubmissionInputDTO} @return the {@link TransactionStatus} @throws FatalException	This method is used to validate the input data present in the given {@link VariationRequestDTO} which represents an Account Variation Request. This method will first validate the account variation request and then validate the submission data associated with the request. @param sessionContext the {@link SessionContext} @param variationRequestDTO the {@link VariationRequestDTO} @return the {@link TransactionStatus} @throws FatalException
IAccountOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountOriginationAdapter	validateSubmission	This service will validate all inputs within the submission. @param sessionContext the {@link SessionException} @param submissionInputDTO the {@link SubmissionInputDTO} @return the	This service will validate all inputs within the submission. @param sessionContext the {@link SessionException} @param submissionInputDTO the {@link SubmissionInputDTO} @return the {@link TransactionStatus} @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			{@link TransactionStatus} @throws FatalException	
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	debitAccount		<p>This method is defined to populate Accounting Event - Amount Tag Details.   It populates AccountingEventAmountTagDetailsDTO. </p> @param applicationContext @param casaAccountId @param amountTag @param amountTagSequenceNumber @param instrumentId @param transAmountInAccountCurrency @return AccountingEventAmountTagDetailsDTO
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	debitAccount		
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	debitAccount		
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	creditAccount		<p>This method is defined to populate Accounting Event - Amount Tag Details.   It populates AccountingEventAmountTagDetailsDTO. </p> @param applicationContext @param casaAccountId @param amountTag @param amountTagSequenceNumber @param instrumentId @param transAmountInAccountCurrency @return AccountingEventAmountTagDetailsDTO
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	creditAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	creditAccount		
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	fetchAccountBasicDetails		<p>This method will call the BranchDDAAdapter to fetch the basic account details for a Demand Deposit Account. </p> @param accountNumber @return @throws FatalException
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	fetchAccountBasicDetails		This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	fetchAccountBasicDetails		
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	fetchAccountBasicDetails		<p>This method is defined to populate Accounting Event - Amount Tag Details.   It populates AccountingEventAmountTagDetailsDTO. </p> @param applicationContext @param casaAccountId @param amountTag @param amountTagSequenceNumber @param instrumentId @param transAmountInAccountCurrency @return AccountingEventAmountTagDetailsDTO
IBranchDDA Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchDDAAdapter	fetchAccountBasicDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	publishRule		
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	publishRule		
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	publishRule		
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	modifyRuleAndNotify		
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	fetchRules		This method exports the Guided Rule object corresponding to given ruleId and effectiveDate into text format rule file @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @param effectiveDate Effective date of Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	fetchRules		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	addExpression		Processes the list of create {@link com.ofss.fc.dsl.expression.expression.create} instance from xtext grammer , converts to expressionDTO and calls the app service to add the expression @param sessionContext @param createRequest @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	addExpression		This represents the create a expression and calls addExpression of ExpressionApplicationService . @param sessionContext @param expressionDTO @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	updateExpression		This represents the update a expression and calls updateExpression of ExpressionApplicationService . @param sessionContext @param expressionDTO @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	updateExpression		Processes the list of update instance from xtext grammer , converts to expressionDTO and calls the app service to update the given expression @param sessionContext @param updateList @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	fetchExpression		This represents the fetch a expression and calls fetchExpression of ExpressionApplicationService @param sessionContext @param expressionDTO @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	fetchExpression		Processes the list of retrieve instance from xtext grammer , converts to expressionDTO and calls the app service to retrieve the expression @param sessionContext @param retrieveList @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getListOfExpressionIDs		This represents fetching a list of expressionIds and calls getListOfExpressionIDs of ExpressionApplicationService . @param sessionContext @param operator @param expName @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getListOfExpressionIDs		Processes the list of retrieve instance from xtext grammer , converts to expressionDTO and calls the app service to retrieve the expression @param sessionContext @param retrieveList @return @throws FatalException
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getOperationDTO		It creates fragmentDTO when expression and arithmetic operator are passed as parameter @param exp @param operator @return FragmentDTO
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getFactCode		
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getFactCode		This method sets the operandDTO @param operandDTO @return
IDSLRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\rule\IDSLRuleAdapter	getEncryptionKey		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceCollectionAdapter	updateBankDetails	@param changeBankWrapperDTO @throws FatalException	@param changeBankWrapperDTO @throws FatalException
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceCollectionAdapter	updateBankDetails	@param changeBankWrapperDTO @throws FatalException	To update the bank details already created in collections module.
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceCollectionAdapter	createBankDetails	@param changeBankWrapperDTO @throws FatalException	@param changeBankWrapperDTO @throws FatalException
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceCollectionAdapter	createBankDetails	@param changeBankWrapperDTO @throws FatalException	To create the bank details in collections module.
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceCollectionAdapter	updateBankBranchDetails	@param changeBankBranchWrapperDTO @throws FatalException	@param changeBankBranchWrapperDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceCollectionAdapter	updateBankBranchDetails	@param changeBankBranchWrapperDTO @throws FatalException	To update the bank branch details already created in collections module.
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceCollectionAdapter	updateBankBranchDetails	@param changeBankBranchWrapperDTO @throws FatalException	To create the bank branch details in collections module.
ICommonServiceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceCollectionAdapter	createBankBranchDetails	@param changeBankBranchWrapperDTO @throws FatalException	@param changeBankBranchWrapperDTO @throws FatalException
IPerformPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPerformPaymentAdapter	inquireAndDoRedemption		
IPerformPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPerformPaymentAdapter	transferFundsToBeneficiaries		
IPerformPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPerformPaymentAdapter	doInstallmentPayment		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPerformPaymentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPerformPaymentAdapter	inquireLoanInstallmentPayment		
IDSLOriginatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLOriginatorAdapter	fetchApplicationBasicInfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	This service retrieves the basic information of the application. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType} @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLOriginatorAdapter	fetchApplicationBasicInfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	
IDSLOriginatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLOriginatorAdapter	fetchApplicationBasicInfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	The following method is used to validate whether the party id belonging to an application context is same as the party id which is already set in party context
IDSLOriginatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLOriginatorAdapter	fetchApplicationBasicInfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	The following method is used to validate whether the application id belongs to submission context



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	dapter		FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationBasicI nfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationBasicI nfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationBasicI nfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	The following method is used to show various account related details .It displays various account related details.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationBasicI nfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationBasicI nfo	This method will fetch basic application information for an application id @param applicationId @return @throws FatalException	The following method is used to set the application in context. Application id is provided by the user in the conversation command.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireSub missionSu mmary	This method will return the summary of submission based on the submission Id @param sessionContext @param submissionID @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application. tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireSub missionSu mmary	This method will return the summary of submission based on the submission Id @param sessionContext @param submissionID @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application. tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	This method is used for fetching all the details associated with the submission Id @param sessionContext @param submissionId @return @throws FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	This method is used to fetch the product groups for incomplete submissions. @param sessionContext @param submissionID @return @throws FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	The following method is used to validate whether the party id belonging to an submission context is same as the party id which is already set in party context
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	The following method is used to show various account related details .It displays various account related details.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub mission	This method will return the DTO which will contain information about a submission including party Id @param sessionContext @param submissionId @return @throws FatalException	The following method is used to set the application in context. Application id is provided by the user in the conversation command.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the application progress. This provides the detail on the application stage, status and the progress. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the application progress needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.application.tracki ng.ApplicationProgressResponse}, This contains the information on application progress details (application stage, status and detailed progress) @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the allowed actions for the application. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the basic information of the application. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the worklist human task items pending for the application @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.workflow.tracker.HumanTaskInquiryResponse}, This contains the information on worklist items / human tasks pending for the application @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service is used to search application / submission summary for the given search criteria input. The method logic is as follows, <ul> <li>Convert ApplicationSearchCriteriaDTO to ApplicationSearchRequestDTO using the builder @link{ApplicationSearchRequestDTOBuilder (invoke the build method)}</li> <li>Search the application by invoking the service, ApplicationSearchApplicationService @link{ApplicationSearchApplicationService#searchA pplication <li>Convert ApplicationSearchResponse to ApplicationSearchSummaryDTO using the builder @link{ApplicationSearchSummaryDTOBuilder by invoking the corresponding build operation.</li> <li>Set the summary DTO in the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.workflow.tracker.HumanTaskIn quiryResponse}, This contains the information on worklist items / human tasks pending for the application @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	This service is used to perform the default search that is a search based on applicationId, submission Id or partyId <ul> <li>@link{com.ofss.fc.app.origination.dto.core.search.i nput.ApplicationSearchRequestDTO} is built using the builder @link{com.ofss.fc.app.origination.service.core.dto.buil der.application.tracking. ApplicationSearchRequestDTOBuilder}</li> <li>ApplicationSearchRequestDTO used to fetch @link{com.ofss.fc.app.origination.dto.core.search.outp ut.BasicApplicationSearchResponse} through the service @link{com.ofss.fc.app.origination.service.core.search.A pplicationSearchApplicationService.performDefaultSe arch(SessionContext, ApplicationSearchRequestDTO ) } < / li > <li> @link{ com . ofss . fc . app . origination . dto . core . application . tracking . ApplicationSearchSummaryDTO } is created using builder @link { com . ofss . fc . app . origination . service . core . dto . builder . application . tracking . BasicApplicationSearchSummaryDTOBuilder . BasicApplicationSearchSummaryDTOBuilder ( ) } </li> <li> @link{ com . ofss . fc . app . origination

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> dto core application tracking ApplicationSearchSummaryResponse } is set and returned &lt;/li&gt; &lt; /ul &gt; @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,@link{com.ofss.fc.app.origination.dto.core.applicati on.tracking.ApplicationSearchCriteriaDTO}, contains the criterion on which applications are to be searched @return @link{com.ofss.fc.app.origination.dto.core.application. tracking.ApplicationSearchSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException </pre>
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationProgre ss	This method is used for fetching the progress of the application associated with the given application Id @param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchPendin gWorklistIt ems	This method will fetch the pending items from the worklist @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the worklist human task items pending for the application @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.workflow.tracker.HumanTaskIn quiryResponse}, This contains the information on worklist items / human tasks pending for the application @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchPendin gWorklistIt ems	This method will fetch the pending items from the worklist @param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchPendin gWorklistIt ems	This method will fetch the pending items from the worklist @param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchAccou ntReferenc e Details	This method is used for fetching the details of opened accounts @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch the account reference details for the given application id. The flow of the method is as follows, <ul> <li>Inquire application details using the service @link{com.ofss.fc.app. origination.service.core.applicati on.BaseApplicationService.inquireApplication </li> <li> Iterate the product group linkage offer reference dto</li> <li> Convert offer reference dto to AccountReferenceDetailDTO using the builder @link{AccountReferenceDetailDTOBuilder by invoking the build operation</li> <li>Set the DTO array in the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,It is the application id, for which the account reference details need to be fetched @return @link{com.ofss.fc.app. origination.dto.core.pg.Account ReferenceDetailReponse}, This contains the array of @link{com.ofss.fc.app. origination.dto.core.pg.Account ReferenceDetailDTO @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchAccou ntReferenc e Details	This method is used for fetching the details of opened accounts @param sessionContext @param applicationId @return @throws FatalException	This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchAccou ntReferenc e Details	This method is used for fetching the details of opened accounts @param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub missionFro mDraft	This method is used for fetching submission draft @param sessionContext @param submissionId @return @throws FatalException	This method is used to fetch the product groups for incomplete submissions. @param sessionContext @param submissionID @return @throws FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub missionFro mDraft	This method is used for fetching submission draft @param sessionContext @param submissionId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub missionFro mDraft	This method is used for fetching submission draft @param sessionContext @param submissionId @return @throws FatalException	The following method is used to show various account related details .It displays various account related details.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireSub missionFro mDraft	This method is used for fetching submission draft @param sessionContext @param submissionId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	searchAppli cation	This method is used to search application based on the criteria passed @param sessionContext @param applicationSearchCriteriaDTO @return ApplicationSearchSummaryResponse @throws FatalException	This service is used to search application / submission summary for the given search criteria input. The method logic is as follows, <ul> <li>Convert ApplicationSearchCriteriaDTO to ApplicationSearchRequestDTO using the builder @link{ApplicationSearchRequestDTOBuilder (invoke the build method)}</li> <li>Search the application by invoking the service, ApplicationSearchApplicationService @link{ApplicationSearchApplicationService#searchApplication} <li>Convert ApplicationSearchResponse to ApplicationSearchSummaryDTO using the builder @link{ApplicationSearchSummaryDTOBuilder} by invoking the corresponding build operation.</li> <li>Set the summary DTO in the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.workflow.tracker.HumanTaskInquiryResponse}, This contains the information on worklist items / human tasks pending for the application @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	searchAppli cation	This method is used to search application based on the criteria passed @param sessionContext @param applicationSearchCriteriaDTO @return ApplicationSearchSummaryResponse @throws FatalException	This service is used to perform the default search that is a search based on applicationId, submission Id or partyId <ul> <li>@link{com.ofss.fc.app.Origination.dto.core.search.input.ApplicationSearchRequestDTO} is built using the builder @link{com.ofss.fc.app.Origination.service.core.dto.builder.ApplicationTrackingApplicationSearchRequestDTOBuilder}</li> <li>ApplicationSearchRequestDTO used to fetch @link{com.ofss.fc.app.Origination.dto.core.search.output.BasicApplicationSearchResponse} through the service @link{com.ofss.fc.app.Origination.service.core.search.ApplicationSearchApplicationService.performDefaultSearch(SessionContext, ApplicationSearchRequestDTO ) } </li> <li> @link{ com . ofss . fc . app . origination . dto . core . application . tracking . ApplicationSearchSummaryDTO } is created using builder @link { com . ofss . fc . app . origination . service . core . dto . builder . application . tracking . BasicApplicationSearchSummaryDTOBuilder . BasicApplicationSearchSummaryDTOBuilder ( ) } </li> <li> @link{ com . ofss . fc . app . origination



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> dto core application tracking ApplicationSearchSummaryResponse } is set and returned &lt;/li&gt; &lt; /ul &gt; @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,@link{com.ofss.fc.app.origination.dto.core.applicati on.tracking.ApplicationSearchCriteriaDTO}, contains the criterion on which applications are to be searched @return @link{com.ofss.fc.app.origination.dto.core.application. tracking.ApplicationSearchSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException </pre>
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	searchAppli cation	This method is used to search application based on the criteria passed @param sessionContext @param applicationSearchCriteriaDTO @return ApplicationSearchSummaryResponse @throws FatalException	This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	searchAppli cation	This method is used to search application based on the criteria passed @param sessionContext @param applicationSearchCriteriaDTO @return ApplicationSearchSummaryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate ralValuatio nSummary	This method is used for fetching the valuation status based upon an application Id @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	This method is used to fetch all collateral valuation summary for the application.   <ul> <li>Fetch application decision using the service @link{CreditDecisionApplicationService#inquireAppli cationDecision </li> <li>Iterate collateral decisions from application decision retrieved above and collect all collateral ids</li> <li>Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCo llateralTitleSearchSummary and get the title search summary </li> <li>Return the above response </li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Application Collateral Valuation Summary needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.application.tracki ng.ApplicationCollateralValuationSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate ralValuatio nSummary	This method is used for fetching the valuation status based upon an application Id @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	This method will fetch the valuation status details and set them in a reponse object. This response will be displayed on the screen in a proper format.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate ralValuatio nSummary	This method is used for fetching the valuation status based upon an application Id @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			ption	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchVerific ations	This method is used for fetching verification details @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch the verification details for the application. The flow of the method is as follows, <ul> <li>Inquire application checklist details using the service @link{com.ofss.fc.domain.origination.service.core.verif ication .ApplicationVerificationDomainService.fetchVerificati onChecklistId }</li> <li>Inquire verification details using the checklist id, by invoking the service @link{com.ofss.fc.app.commonservice .core.service.verification.VerificationApplicationServic e.fetchVerificationChecklistWithHistory }</li> <li>Set the application id and Verification DTO array to the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, It is the application id, for which the verifications need to be fetched @return @link{com.ofss.fc.app.origination.dto.core.verification. ApplicationVerificationResponse}, This contains applicationId and an array of @link{com.ofss.fc.app.commonservice.core.dto.verifica tion.VerificationDTO} @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchVerific ations	This method is used for fetching verification details @param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchVerific ations	This method is used for fetching verification details @param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate ralTitleSear chSummary	This method is used for fetching the title search summary @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	This method is used to fetch all collateral title search summary for the application.   <ul> <li>Fetch application decision using the service <link{CreditDecisionApplicationService#inquireAppli cationDecision </li> <li>Iterate collateral decisions from application decision retrieved above and collect all collateral ids</li> <li>Pass the collateral ids to the service <link{ApplicationTrackingApplicationService#fetchCo llateralValuationSummary and get the valuation summary</li> <li>Return the above response</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc <fcb.param MI,String,applicationId, This object holds the application id, for which the Application Collateral Title Search Summary needs to be retrieved @return <link com.ofss.fc.app.Origination.dto.core.application.tracki ng.ApplicationCollateralTitleSearchSummaryRespons e) @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate ralTitleSear chSummary	This method is used for fetching the title search summary @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationCollate dTitleSear chSummary	This method is used for fetching the title search summary @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationStateTr ansitionHis tory	This method is used for fetching the current state of the application along with the its state change history @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	This service retrieves the Application state transaction history. <ul> <li>Inquire application details using the service </li><li>Inquire state transition history response using the service </li></ul> @link{com.ofss.fc.app. origination.service.core.applicati on.BaseApplicationService.inquireApplication </li> <li>Inquire state transition history response using the service </li> @link{StateTransitionApplicationService#inquireState TransitionHistory . Pass application id as entity id and entity type as OriginationConstants.LENDING_APPLICATION / CREDIT_CARD_APPLICATION / DEPOSIT_APPLICATION, based on the application type</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc <li>@fcb.param MI,String,applicationId, This object holds the application id, for which the Application State Transition History needs to be retrieved @return <li>@link com.ofss.fc.app.commonservice.core.dto.statetransitio n.StateTransitionHistoryResponse}, This contains the information on state transition history details(StateTransitionHistoryDTO and currentState) <li>@throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationStateTr ansitionHis tory	This method is used for fetching the current state of the application along with the its state change history @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationStateTr ansitionHis tory	This method is used for fetching the current state of the application along with the its state change history @param sessionContext @param applicationId @return @throws com.ofss.fc.infra.exception.FatalExce ption	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationComm ents	@param sessionContext @param applicationId @return @throws FatalException	This method is to fetch all application level comments and submission level primary linkage comments.  <ul> <li>Fetch comment inquiry response using the service @link{CommentLinkageApplicationService.fetchCommentsBySearchType(SessionContext, CommentSearchDTO[])}. The input of the service is constructed as follows, <li>Since comments needs to be fetch for both application and submission at different context, construct two comment search DTO as follows</li> <li>Comment search DTO 1: CommentSearchType = FETCH_ALL, EntityCommentLinkageDTO with linkedEntityKey as application id</li> <li>Comment search DTO 2: CommentSearchType = FETCH_PRIMARY, EntityCommentLinkageDTO with linkedEntityKey as submission id. Submission id can be fetched from ApplicationDTO by invoking the service

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@link{BaseApplicationService.inquireApplication  &lt;/li&gt; &lt;/li&gt; &lt;/ul&gt; @fcb.param  MI,SessionContext,sessionContext, ,Session Context  which must contain the session context information  such as user id, branch, branch code, channel etc  @fcb.param MI,String,applicationId, ,This object holds  the application id, for which the Application  Comments needs to be retrieved @return {@link  com.ofss.fc.app.commonservice.core.inquiryresponse.  CommentInquiryResponse} this class contains the  array of CommentDTOs. @throws  com.ofss.fc.infra.exception.FatalException</p>
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationComm ents	@param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchApplic ationComm ents	@param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	fetchSubmi ssionComm ents		<p>This method is to fetch all submission level primary linkage comments. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Fetch comment inquiry response using the service @link{CommentLinkageApplicationService.fetchCommentsBySearchType(SessionContext, CommentSearchDTO[])}. The input of the service is constructed as follows, &lt;li&gt;Since comments needs to be fetch for both application and submission at different context, construct two comment search DTO as follows&lt;/li&gt; &lt;li&gt;Comment search DTO 1: CommentSearchType = FETCH_ALL, EntityCommentLinkageDTO with linkedEntityKey as application id&lt;/li&gt; &lt;li&gt;Comment search DTO 1: CommentSearchType = FETCH_PRIMARY, EntityCommentLinkageDTO with linkedEntityKey as submission id. Submission id can be fetched from ApplicationDTO by invoking the service @link{BaseApplicationService.inquireApplication} &lt;/li&gt; &lt;/li&gt; &lt;/ul&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, This object holds the submission id, for which the Submission Comments needs to be retrieved @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentInquiryResponse} this class contains the array of CommentDTOs. @throws com.ofss.fc.infra.exception.FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchSubmi ssionComm ents		This method is used for fetching all the details associated with the submission Id @param sessionContext @param submissionId @return @throws FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchSubmi ssionComm ents		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	fetchSubmi ssionComm ents		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	addUserCo mmentForS ubmission		This method is used to add user comment for the application for submission level entered from application tracker.   <ul> <li>Add user comment by invoking the service @link{CommentLinkageApplicationService.addOrModifyComments(SessionContext, CommentDTO[], String)}. The input of the service is constructed or populated as below,  <li>Set current stage to "Application Tracker"</li> <li>Construct CommentDTO, using subject, comment text, isInternalToBank and application id as entity linkage (primary) to it. Set CommentSourceType to User. Stage as above.</li> </li> <li>Retrieve CommentDTO from response @link{AddCommentResponse above retrieved}</li> <li>Set the CommentDTO to response @link{CommentResponse}</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@fcb.param MI,String,submissionId, ,This object holds the submission id, for the user comment that has to be added @fcb.param MI ,String,subject, ,This object holds the subject, for the user comment that has to be added @fcb.param MI ,String,commentText, , This object holds the comment text, for the user comment that has to be added @fcb.param MI ,Boolean,isInternalToBank, , This object holds the is internal to bank,for the user comment that has to be added @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentResponse} @throws com.ofss.fc.infra.exception.FatalException</p>
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mmentForS ubmission		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mmentForS ubmission		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mment		<p>This method is used to add user comment for the application entered from application tracker. &lt;br&gt;</p> <ul style="list-style-type: none"> <li>Add user comment by invoking the service @link{CommentLinkageApplicationService.addOrModifyComments(SessionContext, CommentDTO[], String)}. The input of the service is constructed or populated as below,&lt;br&gt;</li> <li>Set current stage to "Application Tracker"&lt;/li&gt;</li> <li>Construct CommentDTO, using subject, comment text, isInternalToBank and application id as entity linkage (primary) to it. Set CommentSourceType to User. Stage as above.&lt;/li&gt;</li> <li>Retrieve CommentDTO from response @link{AddCommentResponse above retrieved}&lt;/li&gt;</li> <li>Set the CommentDTO to response @link{CommentResponse}&lt;/li&gt;</li> </ul> <p>@fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for the user comment that has to be added @fcb.param MI,String,subject, This object holds the subject, for the user comment that has to be added @fcb.param MI,String,commentText, This object holds the comment text, for the user comment that has to be added @fcb.param MI,Boolean,isInternalToBank, This object holds the is internal to bank,for the user comment that has to be added @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mment		<p>This method is used to add user comment for the application for submission level entered from application tracker. &lt;br&gt; &lt;ul&gt; &lt;li&gt;Add user comment by invoking the service @link{CommentLinkageApplicationService.addOrModifyComments(SessionContext, CommentDTO[], String)}. The input of the service is constructed or populated as below,&lt;br&gt; &lt;li&gt;Set current stage to "Application Tracker"&lt;/li&gt; &lt;li&gt;Construct CommentDTO, using subject, comment text, isInternalToBank and application id as entity linkage (primary) to it. Set CommentSourceType to User. Stage as above.&lt;/li&gt; &lt;/li&gt; &lt;li&gt;Retrieve CommentDTO from response @link{AddCommentResponse above retrieved}&lt;/li&gt; &lt;li&gt;Set the CommentDTO to response @link{CommentResponse}&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, This object holds the submission id, for the user comment that has to be added @fcb.param MI,String,subject, This object holds the subject, for the user comment that has to be added @fcb.param MI,String,commentText, This object holds the comment text, for the user comment that has to be added @fcb.param MI,Boolean,isInternalToBank, This object holds the is internal to bank,for the user comment that has to be added @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mment		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mment		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	addUserCo mment		
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the application progress. This provides the detail on the application stage, status and the progress. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.application.tracki ng.ApplicationProgressResponse}, This contains the information on application progress details (application stage, status and detailed progress) @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the allowed actions for the application. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app. origination.dto.core.application. tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the basic information of the application. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app. origination.dto.core.application. tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.Application TrackingActionType @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the submission status using the application id @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.application.tracking.ActionInquiryReponse}, This contains the information on actions allowed for the application. Represented as enum @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This service retrieves the worklist human task items pending for the application @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the application progress needs to be retrieved @return @link{com.ofss.fc.app.workflow.tracker.HumanTaskInquiryResponse}, This contains the information on worklist items / human tasks pending for the application @link{com.ofss.fc.enumeration.origination.ApplicationTrackingActionType @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch the account reference details for the given application id. The flow of the method is as follows, <ul> <li>Inquire application details using the service @link{com.ofss.fc.app.origination.service.core.applicati on.BaseApplicationService.inquireApplication </li> <li> Iterate the product group linkage offer reference dto</li> <li> Convert offer reference dto to AccountReferenceDetailDTO using the builder @link{AccountReferenceDetailDTOBuilder by invoking the build operation</li> <li>Set the DTO array in the response object</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,It is the application id, for which the account reference details need to be fetched @return @link{com.ofss.fc.app.origination.dto.core.pg.Account ReferenceDetailReponse}, This contains the array of @link{com.ofss.fc.app.origination.dto.core.pg.Account ReferenceDetailDTO @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch the verification details for the application. The flow of the method is as follows, <ul style="list-style-type: none"> <li>Inquire application checklist details using the service @link{com.ofss.fc.domain.origination.service.core.verif ication .ApplicationVerificationDomainService.fetchVerificati onChecklistId }&lt;/li&gt; <li>Inquire verification details using the checklist id, by invoking the service @link{com.ofss.fc.app.commonservice .core.service.verification.VerificationApplicationServic e.fetchVerificationChecklistWithHistory }&lt;/li&gt; <li>Set the application id and Verification DTO array to the response object&lt;/li&gt; </li></li></li></ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, It is the application id, for which the verifications need to be fetched @return @link{com.ofss.fc.app.origination.dto.core.verification. ApplicationVerificationResponse}, This contains applicationId and an array of @link{com.ofss.fc.app.commonservice.core.dto.verifica tion.VerificationDTO} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch mortgage insurance summary for the application id and facility.   <ul> <li>Inquire application details for the application id input, by invoking the service @link{BaseApplicationService#inquireApplication </li> <li>Fetch product group serial number for the facility id input from the retrieved application details above</li> <li>Fetch LenderRiskMgmtDetails by constructed the key using the application id and product group serial number and using the service @link{LenderRiskMgmtApplicationService#fetchLRM Policy}</li> <li>Retrieve LenderRiskMgmtDetailsDTO from the response above, Convert LenderRiskMgmtDetailsDTO to MortgageInsuranceSummaryDTO using the builder @link{MortgageInsuranceSummaryDTOBuilder#build } </li> <li>Set MortgageInsuranceSummaryDTO to MortgageInsuranceSummaryResponse</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Mortgage Insurance Summary needs to be retrieved @fcb.param MI,String,facilityId, ,This object holds the facility id for which the Mortgage Insurance Summary needs to be retrieved @return {@link com.ofss.fc.app.Origination.dto.lending.core.lrm.MortgageInsuranceSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com \ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch mortgage insurance details for the application id and facility id.   <ul> <li>Inquire application details for the application id input, by invoking the service @link{BaseApplicationService#inquireApplication </li> <li>Fetch product group serial number for the facility id input from the retrieved application details above</li> <li>Fetch LenderRiskMgmtDetails by constructed the key using the application id and product group serial number and using the service @link{LenderRiskMgmtApplicationService#fetchLRM Policy}</li> <li>Retrieve LenderRiskMgmtDetailsDTO from the response above, Convert LenderRiskMgmtDetailsDTO to MortgageInsuranceDetailDTO using the builder @link{MortgageInsuranceDetailDTOBuilder#build}</li> <li>Set MortgageInsuranceDetailDTO to MortgageInsuranceDetailResponse</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Mortgage Insurance Details needs to be retrieved @fcb.param MI,String,facilityId, ,This object holds the facility id for which the Mortgage Insurance Details needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.lending.core.lrm.MortgageInsuranceDetailResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch mortgage insurance waiver details for the application id and facility id.   <ul> <li>Inquire application details for the application id input, by invoking the service @link{BaseApplicationService#inquireApplication </li> <li>Fetch product group serial number for the facility id input from the retrieved application details above</li> <li>Fetch LenderRiskMgmtDetails by constructed the key using the application id and product group serial number and using the service @link{LenderRiskMgmtApplicationService#fetchLRM Policy}</li> <li>Retrieve LenderRiskMgmtWaiverDetailsDTO from LenderRiskMgmtDetailsDTO of the response retrieved above. Convert LenderRiskMgmtWaiverDetailsDTO to MortgageInsuranceWaiverDTO using the builder @link{MortgageInsuranceWaiverDTOBuilder#build </li> <li>Set MortgageInsuranceWaiverDTO to MortgageInsuranceWaiverResponse</li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Mortgage Insurance Waiver Details needs to be retrieved @fcb.param MI,String,facilityId, ,This object holds the facility id for which the Mortgage Insurance Waiver Details needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.lending.core.lrm.MortgageInsuranceWaiverResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is to fetch all application level comments and submission level primary linkage comments.  <ul> <li>Fetch comment inquiry response using the service @link{CommentLinkageApplicationService.fetchCommentsBySearchType(SessionContext, CommentSearchDTO[])}. The input of the service is constructed as follows, <li>Since comments needs to be fetch for both application and submission at different context, construct two comment search DTO as follows</li> <li>Comment search DTO 1: CommentSearchType = FETCH_ALL, EntityCommentLinkageDTO with linkedEntityKey as application id</li> <li>Comment search DTO 1: CommentSearchType = FETCH_PRIMARY, EntityCommentLinkageDTO with linkedEntityKey as submission id. Submission id can be fetched from ApplicationDTO by invoking the service @link{BaseApplicationService.inquireApplication} </li> </li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the Application Comments needs to be retrieved @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentInquiryResponse} this class contains the array of CommentDTOs. @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to add user comment for the application entered from application tracker.  <ul> <li>Add user comment by invoking the service @link{CommentLinkageApplicationService.addOrModifyComments(SessionContext, CommentDTO[], String)}. The input of the service is constructed or populated as below,  <li>Set current stage to "Application Tracker"</li> <li>Construct CommentDTO, using subject, comment text, isInternalToBank and application id as entity linkage (primary) to it. Set CommentSourceType to User. Stage as above.</li> </li> <li>Retrieve CommentDTO from response @link{AddCommentResponse above retrieved}</li> <li>Set the CommentDTO to response @link{CommentResponse}</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for the user comment that has to be added @fcb.param MI,String,subject, This object holds the subject, for the user comment that has to be added @fcb.param MI,String,commentText, This object holds the comment text, for the user comment that has to be added @fcb.param MI,Boolean,isInternalToBank, This object holds the is internal to bank,for the user comment that has to be added @return {@link com.ofss.fc.app.commonservice.core.inquiryresponse.CommentResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch all collateral valuation summary for the application.   <ul> <li>Fetch application decision using the service @link{CreditDecisionApplicationService#inquireApplicationDecision </li> <li>Iterate collateral decisions from application decision retrieved above and collect all collateral ids</li> <li>Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCollateralTitleSearchSummary and get the title search summary </li> <li>Return the above response </li> </ul> @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, ,This object holds the application id, for which the Application Collateral Valuation Summary needs to be retrieved @return {@link com.ofss.fc.app. origination.dto.core.application.tracki ng.ApplicationCollateralValuationSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch all collateral title search summary for the application.   <ul> <li>Fetch application decision using the service @link{CreditDecisionApplicationService#inquireApplicationDecision </li> <li>Iterate collateral decisions from application decision retrieved above and collect all collateral ids</li> <li>Pass the collateral ids to the service @link{ApplicationTrackingApplicationService#fetchCollateralValuationSummary and get the valuation summary</li> <li>Return the above response</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the Application Collateral Title Search Summary needs to be retrieved @return {@link com.ofss.fc.app.origination.dto.core.application.tracking.ApplicationCollateralTitleSearchSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	Utility method to get the application type @param applicationId @return String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to fetch origination fees using the application id and facility id. The logic of the method is as follows</br> <ol> <li>Fetch pricing ticket number using the service @link{OriginationFeeApplicationService#fetchOriginationFees(SessionContext, HeaderDTO, LendingProductGroupLinkageDTO, ApplicantDataDTO, CreditFacilityDTO[], CollateralDTO[]).</br> The input of This service can be derived as follows,</br> <ul> <li>SessionContext - SessionContext from the input HeaderDTO - Inquire application details using base application service @link{com.ofss.fc.app.origination.service.core.application.BaseApplicationService.inquireApplication} using the application id, input. Pass the header dto from the ApplicationDTO retrieved.</li> <li>LendingProductGroupLinkageDTO - Fetch product group linkage from the ApplicationDTO retrieved above, using the facility id, input. Set the PricingTicketNumberForBundleFees to the pg linkage dto, if IsNominatedForBundle() and PricingTicketNumberForBundleFees() is empty, by getting the new pricing ticket number by invoking the service @link{OriginationFeeApplicationService#createBundleSubscriptionPricingPlan </li> <li>ApplicantDataDTO <li>Fetch all applicants for the application using the helper method @link{LendingApplicationHelper#fetchAllApplicantsForApplicationId </li> <li>Convert applicant DTO To applicant data DTO using the helper method @link{SubmissionInquiryHelper#populateApplicantDataDTO </li></li> <li>CreditFacilityDTO[] - Fetch credit facility array DTO using the service

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> @link{LendingSimulationApplicationService#fetchFacilityHierarchy &lt;/li&gt; &lt;li&gt;CollateralDTO[] - Fetch collateral using the service @link{LendingApplicationService#fetchCollateralsForApplication}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Use the pricing ticket number and get the pricing details using the service @link{OriginationPricingAdapter#fetchFeesForTicketNumber} . The input facts can be populated using the methods @link{LendingApplicationService#populateFactsForOriginationFee} and @link{LendingApplicationService#populateFundingFactDefaultValues &lt;/li&gt; &lt;li&gt;Iterate WrappedPriceFeatureInquiryResponse #wrappedPriceFeatureDTO[], convert the WrappedPriceFeatureDTO to PricingFeeDetailDTO using the builder @link{PricingFeeDetailDTOBuilder#build &lt;/li&gt; &lt;li&gt;Set the pricing ticket number and PricingFeeDetailDTOs in the response&lt;/li&gt; &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,applicationId, This object holds the application id, for which the origination fees is to be retrieved @fcb.param MI,String,facilityId, This object holds the facility id, for which the origination fees is to be retrieved @return @link{com.ofss.fc.app.origination.dto.core.pricing.PricingFeesResponse}, This contains the information on origination fees (pricingTicketNumber and pricingFeeDetails) @throws com.ofss.fc.infra.exception.FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com \ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method is used to process origination fees using input ProcessOriginationFeesRequestDTO. ProcessOriginationFeesRequestDTO contains information: application id, facility id, PricingFeeDetailDTO and SettlementDetailsDTO. The logic of the method is as follows  <ul> <li>Iterate PricingFeeDetailDTO array from ProcessOriginationFeesRequestDTO and convert PricingFeeDetailDTO to WrappedPriceFeatureDTO using the builder operation @link{WrappedPriceFeatureDTOBuilder#build}</li> <li>Covert SettlementDetailsDTO to ComponentPreferenceDTO using the builder operation @link{ComponentPreferenceDTO#build}</li> <li>Derive applicant data dto by <li>Fetch all applicants for the application using the helper method @link{LendingApplicationHelper#fetchAllApplicantsForApplicationId}</li> <li>Convert applicant DTO To applicant data DTO using the helper method @link{SubmissionInquiryHelper#populateApplicantDataDTO}</li> </li> <li>Process origination fees using the service @link{OriginationFeeApplicationService.processFees}</li> </ul> @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,@link{com.ofss.fc.app.Origination.dto.core.pricing.ProcessOriginationFeesRequestDTO },processOriginationFeesRequestDTO, This object holds the application id,facility id,settlement details and an array of pricing fee details for which the origination fees needs to be processed @return @link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.service.response.TransactionStatus}This entity holds information on the transaction status. @throws com.ofss.fc.infra.exception.FatalException
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	This method will fetch the valuation status details and set them in a reponse object. This response will be displayed on the screen in a proper format.
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionA dapter	inquireApp lication	This method is used to fetch the information related to the given applicaion Id @param sessionContext @param applicationId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com \ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	performDef aultSearch	This method is used to basic application search. @param sessionContext @param applicationSearchCriteriaDTO @return @throws FatalException	This service is used to perform the default search that is a search based on applicationId, submission Id or partyId <ul> <li>@link{com.ofss.fc.app.origination.dto.core.search.i nput.ApplicationSearchRequestDTO} is built using the builder @link{com.ofss.fc.app.origination.service.core.dto.buil der.application.tracking. ApplicationSearchRequestDTOBuilder}</li> <li>ApplicationSearchRequestDTO used to fetch @link{com.ofss.fc.app.origination.dto.core.search.outp ut.BasicApplicationSearchResponse} through the service @link{com.ofss.fc.app.origination.service.core.search.A pplicationSearchApplicationService.performDefaultSe arch(SessionContext, ApplicationSearchRequestDTO ) } < / li > < li > @link{ com . ofss . fc . app . origination . dto . core . application . tracking . ApplicationSearchSummaryDTO } is created using builder @link { com . ofss . fc . app . origination . service . core . dto . builder . application . tracking . BasicApplicationSearchSummaryDTOBuilder . BasicApplicationSearchSummaryDTOBuilder ( ) } < / li > < li > @link{ com . ofss . fc . app . origination

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> dto core application tracking ApplicationSearchSummaryResponse } is set and returned &lt;/li&gt; &lt; / ul &gt; @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,@link{com.ofss.fc.app.origination.dto.core.applicati on.tracking.ApplicationSearchCriteriaDTO}, contains the criterion on which applications are to be searched @return @link{com.ofss.fc.app.origination.dto.core.application. tracking.ApplicationSearchSummaryResponse} @throws com.ofss.fc.infra.exception.FatalException </pre>
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	performDef aultSearch	This method is used to basic application search. @param sessionContext @param applicationSearchCriteriaDTO @return @throws FatalException	This method will fetch the details of the opened accounts and set them in the response text which will be displayed on the screen in the desired format @param responseText
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	performDef aultSearch	This method is used to basic application search. @param sessionContext @param applicationSearchCriteriaDTO @return @throws FatalException	
IDSLOriginat ionAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dsl\IDSLOriginat ionAdapter	performDef aultSearch	This method is used to basic application search. @param sessionContext @param applicationSearchCriteriaDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationProcessAdapter	callValuationClientProcessForCollateralUpdate		
ICollateralValuationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationProcessAdapter	callValuationClientProcessForCollateralUpdate		This method calls the functionality to check and update market value. This method has been exposed to be called from the process when a particular collateral valuation request has been completed.  The check needs to be done if the current valuation request is the last pending request and all other requests have been completed or cancelled.  Collateral market value update will only be done when all the requests for a valuation are <b><i>completed or cancelled.</i></b>  @param SessionContext aSessionContext : This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param String valuationReferenceNumber : This parameter specifies the valuation reference number of the valuation. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of TransactionStatus contains the information whether the transaction is success\failure.  @throws FatalException
ILoanRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanRuleAdapter	executeRuleForRegulatedLoan	This method executes the Rule to determine if loan account is Regulated/Non Regulated. @param approvedAmount @param productCode @param offerId @param partyType @param loanPurposeType @return	@param requestedBaseAmount @param productCode @param offerId @param partyType @param loanPurposeType @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			isLoanRegulated	
ILoanRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanRuleAdapter	executeRuleForRegulatedLoan	This method executes the Rule to determine if loan account is Regulated/Non Regulated. @param approvedAmount @param productCode @param offerId @param partyType @param loanPurposeType @return isLoanRegulated	This method executes the Rule to determine if loan account is Regulated/Non Regulated. @param requestedBaseAmount @param productCode @param offerId @param partyType @param loanPurposeType @return isLoanRegulated
ITaxationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationAccountingAdapter	raiseAccountingEvent	This method is used to raise the accounting event. @param accountingRequest @throws FatalException	This method returns object of adapter class
ITaxationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationAccountingAdapter	raiseAccountingEvent	This method is used to raise the accounting event. @param accountingRequest @throws FatalException	This method will raise Accounting Event that is related to taxation Processing. The method will internally delegate the call to {@link TaxationAccountingAdapter} which will internally route the call to accounting module. This method accepts {@link AccountingRequest} object as input which contains details of accounting event. Following is the algorithm for this method:  <ol> <li>Populate {@link TaxationAccountingEvent} object using the details provided in {@link AccountingRequest} object.</li> <li>Get concrete instance of {@link ITaxationAccountingAdapter} object using Adapter Factory.</li> <li>Call raiseAccountingEvent method of {@link ITaxationAccountingAdapter} to raise accounting event. Send {@link TaxationAccountingEvent} object as input to this method.</li> <ol> @param accountingRequest {@link AccountingRequest} object containing details of accounting event.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationAccountingAdapter	raiseAccountingEvent	This method is used to raise the accounting event. @param accountingRequest @throws FatalException	This method will raise Accounting Event that is related to taxation Processing. The method will internally delegate the call to {@link TaxationAccountingAdapter} which will internally route the call to accounting module. This method accepts {@link AccountingRequest} object as input which contains details of accounting event. Following is the algorithm for this method:  <ol> <li>Populate {@link TaxationAccountingEvent} object using the details provided in {@link AccountingRequest} object.</li> <li>Get concrete instance of {@link ITaxationAccountingAdapter} object using Adapter Factory.</li> <li>Call raiseAccountingEvent method of {@link ITaxationAccountingAdapter} to raise accounting event. Send {@link TaxationAccountingEvent} object as input to this method.</li> <ol> @param accountingRequest {@link AccountingRequest} object containing details of accounting event.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationAccountingAdapter	fetchGICode		This method will raise Accounting Event that is related to taxation Processing. The method will internally delegate the call to {@link TaxationAccountingAdapter} which will internally route the call to accounting module. This method accepts {@link AccountingRequest} object as input which contains details of accounting event. Following is the algorithm for this method:  <ol> <li>Populate {@link TaxationAccountingEvent} object using the details provided in {@link AccountingRequest} object.</li> <li>Get concrete instance of {@link ITaxationAccountingAdapter} object using Adapter Factory.</li> <li>Call raiseAccountingEvent method of {@link ITaxationAccountingAdapter} to raise accounting event. Send {@link TaxationAccountingEvent} object as input to this method.</li> <ol> @param accountingRequest {@link AccountingRequest} object containing details of accounting event.
ITaxationAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationAccountingAdapter	fetchGICode		This method is used to fetch the ledger code by calling PMU service. @param taxationAccountingEvent @return @throws FatalException
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the party kyc update operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link IMDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the account closure operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link MDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param accountId
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the account closure operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link MDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param accountId
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the account closure operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link MDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param accountId
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method takes application context, accountId, accountType and operation and logs  record in the table for publishing to MDM. It populates MDMBatchIntegrationDTO using the  information from {@link ApplicationContext} and rest of the parameters and invokes logMDMBatchIntegrationData() to log in the table. @param applicationContext @param accountId @param accountType @param operation

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the account closure operation iff MDM integration is enabled. This method fetches {@link IMDMBatchIntegrationAdapter} and populates {@link IMDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param accountId
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method is invoked after the account writeoff operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link IMDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param accountId
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationAccountData	This method is invoked from account action executors to log the entity in db. @param applicationContext @param accountId @param accountType @param operation @throws FatalException	This method fetches adapter for logging account information during batch execution. @param applicationContext @param demandDepositAccount
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationPartyData	This method is invoked from party action executors to log the entity in db @param applicationContext @param partyId @param operation @throws FatalException	This method is invoked after the party kyc update operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link IMDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param applicationContext @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMBatchIntegrationPartyData	This method is invoked from party action executors to log the entity in db @param applicationContext @param partyId @param operation @throws FatalException	This method takes application context, partyId and operation and logs  record in the table for publishing to MDM. It populates MDMMBatchIntegrationDTO   using the information from {@link ApplicationContext}  and rest of the parameters and invokes logMDMBatchIntegrationData() to log in the table. @param applicationContext @param partyId @param operation
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	logMDMOnlineIntegrationData	This method logs the XML request in the database @param sessioncontext @param xmlRequest @param transactionMode	This method fetches {@link IMDMBatchIntegrationAdapter} using {@link IntegrationAdapterFactory}  and logs the mdm request xml in database which is the published to MDM. @param message
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	fetchMDMOnlineIntegrationData	This method fetches the domain entity using the UUID stored in the thread. @param sessioncontext @param keyId @return	This method fetches the XML request using the UUID set in {@link IntegrationThreadAttribute}   with help of {@link IMDMBatchIntegrationAdapter} and invokes {@link JMSProtocol} to publish the request.  Post publish it updates the status of the record as SUCCESS or FAILED in the database and clears the thread attribute.
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	fetchMDMOnlineIntegrationData	This method fetches the domain entity using the UUID stored in the thread. @param sessioncontext @param keyId @return	This method fetches and return the record from database using keyid
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	fetchMDMOnlineIntegrationData	This method fetches the domain entity using the UUID stored in the thread. @param sessioncontext @param keyId @return	This method fetches the XML request using the UUID set in {@link IntegrationThreadAttribute}   with help of {@link IMDMBatchIntegrationAdapter} and invokes {@link JMSProtocol} to publish the request.  Post publish it updates the status of the record as SUCCESS or FAILED in the database and clears the thread attribute.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	updateMDMOnlineIntegrationData	This method updates the status of the entity post publish @param mdmBatchIntegrationDTO @param status	This method fetches the XML request using the UUID set in {@link IntegrationThreadAttribute}   with help of {@link IMDMBatchIntegrationAdapter} and invokes {@link JMSProtocol} to publish the request.  Post publish it updates the status of the record as SUCCESS or FAILED in the database and clears the thread attribute.
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	updateMDMOnlineIntegrationData	This method updates the status of the entity post publish @param mdmBatchIntegrationDTO @param status	This method updates the record status in database post publish to MDM.
IMDMBatchIntegrationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationAdapter	updateMDMOnlineIntegrationData	This method updates the status of the entity post publish @param mdmBatchIntegrationDTO @param status	This method fetches the XML request using the UUID set in {@link IntegrationThreadAttribute}   with help of {@link IMDMBatchIntegrationAdapter} and invokes {@link JMSProtocol} to publish the request.  Post publish it updates the status of the record as SUCCESS or FAILED in the database and clears the thread attribute.
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<p>This method will check if the supplied nomination preference is null. If the supplied value is null, it will call settlement to create a nomination preference, else it will call settlement to update the existing. </p> @param newBundleAccountLinkage @param nominationPreference @param componentPreferenceDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<pre>This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in Account module. This method will be called in the context of an Instruction setup in Account where in there is going be periodic transfer of funds from a Loan Account/Casa Account to transaction accounts or vice versa </pre> @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	In case the maturity date falls on a holiday, it gets picked up for maturity processing on the next working day. In such a case whether the interest should be given for the number of days lapsed between the actual maturity date and the date on which the deposit gets actually picked up for maturity processing needs to be decided. The decision is made on the following : <ul> <li>The product level flag for additional interest on maturity is true. <li>P+I instruction is maintained with the following settlement setup : <ul> <li>The settlement split type is PERCENT and split value is 100 (i.e single settlement mode) <li>The settlement mode is NOT any flavor of CASA, TD or Loan account </ul> </ul> @param deposit @param dueDateFromAction @return
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method Determines the value date for logging account transaction events on deposit maturity. The date is determined based on the Payout settlement mode. @param deposit @param valueDate @param payoutInstructionType @return Value Date
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method will be called for fetching the component preference DTOs maintained for contribution, disbursement and settlement payout. Fetch method of settlement payout will internally call this method for fetching the componentPreferenceDTO maintained for all the instructions. This will then call inquireComponentPreferenceDetails() method for fetching the details for that instruction and account Id through account settlement adapter. @param sessionContext @param settlementPayoutInstructionDTO @throws FatalException
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	<p>This method will delete the nomination preference when the bundle is being unsubscribed or when the bundle expires. </p> @param bundleReferenceId @param nominationPreference

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	<p>This method will check if the supplied nomination preference is null. If the supplied value is null, it will call settlement to create a nomination preference, else it will call settlement to update the existing. </p> @param newBundleAccountLinkage @param nominationPreference @param componentPreferenceDTO @return
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method delete the settlement payout instruction maintained for a account, facility or application level . @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method closes the Principal, PrincipalInterest and Interest Instructions @param maturityProcessInformation
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	fetchByVersion	This method fetches the ComponentPreferenceDTO stored against the reference number and the accountId by its version. @param sessionContext @param componentPreferenceDTO @return	This method fetch the ComponentPreferenceDTO by its version. @param componentPreferenceDto
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method returns the {@link IAccountSettlementAdapter} to enable interaction between the account and settlement module. @param adapterConstant @return IAccountSettlementAdapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param sessionContext @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline} This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.   @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method executes the settlement payout instruction and raises an accounting event for the same.   It does the following,    <ol> <li>It loads {@link SettlementPayoutItem} via {@link SettlementPayoutInstructionItemsRepository} to obtain settlement reference number (i.e componentPreferenceId)</li> <li>Executes Settlement Instruction via {@link IAccountSettlementAdapter#executePreference} method.</li> <li>If execution of settlement instruction is successful (settlementExecutionResponse.getSummary().getUnsettledAmount().isZero()), it raises an accounting event via {@link SettlementInstructionPayoutApplication#raiseAccounting} method else thrown a {@link BusinessException}</li> </ol> @param applicationContext @param instructionId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.   @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			moduleReferenceNo @param messageType @throws FatalException	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param sessionContext @param eventCode @param narration
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param sessionContext @param eventCode @param narration
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			narration	
IAccountSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountSettlementAdapter	reverseUsingComponentPreferenceId	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, SettlementExecutionService. @param eventCode @param narration	
ICommonServiceVerificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceVerificationAdapter	fetchEventName	This method fetch the EventName from the EventApplicationService for the given eventId. @param sessionContext,eventId @return String	This method is used to set all the additional fields that doesn't get fetched with hibernate query. @param domainObjectDTOArray @throws FatalException
ICommonServiceVerificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceVerificationAdapter	fetchEventName	This method fetch the EventName from the EventApplicationService for the given eventId. @param sessionContext,eventId @return String	This method will inquire a verification policy record. A verification policy has 2 elements as   part of the key elements i.e the policy identifier and the effective date combination.   One verification policy can have 1 or more effective date records maintained and active.   The rules service is called to fetch the rule definition.  Based on the policy Id and effective date inputs from the user the corresponding verification policy   will be fetched.If a record for the combination of verification policy   identifier and effective date does not exist, then the system will throw an exception.   @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceVerificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons\service\ICommonServiceVerificationAdapter	fetchEventName	This method fetch the EventName from the EventApplicationService for the given eventId. @param sessionId, eventId @return String	This method is used to version requested by the user for an entity. @param sessionId : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IOriginationDebitCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDebitCardAdapter	originateDebitCard	This method is used for setting up a debit card for an applicant in Origination @param MI, SessionContext, sessionId, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @param MI, DebitCardRequestDTO, debitCardRequestDTO, , Indicates information for account user, account, card, currency code, etc. @return DebitCardOriginationResponse, Indicates response containing the debit card number, the currency and the current debit card status. @throws FatalException	@return Object

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDebitCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDebitCardAdapter	originateDebitCard	This method is used for setting up a debit card for an applicant in Origination @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, DebitCardRequestDTO, debitCardRequestDTO, , Indicates information for account user, account, card, currency code, etc. @return DebitCardOriginationResponse, Indicates response containing the debit card number, the currency and the current debit card status. @throws FatalException	The method originateDebitCard is responsible for actually doing debit card request   The actual implementation of debit card will be based on runtime configuration and will be done via adapter methodology   @param sessionContext @param debitCardRequestDTO @return TransactionStatus @throws FatalException
IOriginationDebitCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDebitCardAdapter	inquireCardAccountLinkagesForParty	This method fetches all the existing cards with all account linkages for given party @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the applicant in party module. @return CardSlotLinkagesResponseDTO, Indicates response containing card slot information for an existing party. @throws FatalException	@return Object

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationDebitCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDebitCardAdapter	inquireCardAccountLinkagesForParty	This method fetches all the existing cards with all account linkages for given party @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI, String, partyId, , Indicates the identifier for the applicant in party module. @return CardSlotLinkagesResponseDTO, Indicates response containing card slot information for an existing party. @throws FatalException	
IOriginationDebitCardAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationDebitCardAdapter	fetchPrefixDescOfParty	This method fetch the prefix description of a Party for prefix id. @fcb.param MI, String prefixId, Id of the prefix for particular Party. @return String, Indicate the prefix description @throws FatalException	@return Object
IAccountWriteOffProcessRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\IAccountWriteOffProcessRequestAdapter	initiateAccountWriteOffProcess		@see com.ofss.fc.app.collection.service.accountwriteoffprocess .IAccountWriteOffProcessApplicationService This method creates case from accountNumber. @param p_SessionContext @param accountNumber @throws FatalException
IAccountWriteOffProcessRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\IAccountWriteOffProcessRequestAdapter	initiateAccountWriteOffProcess		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOpsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IOpsPricingAdapter	validatePricingPollerStatus	Validates Pricing Poller status for EOD start @param processDate @param nextProcessDate @param branchGroupCode @throws FatalException	This method will validate the re-pricing category status. The validation is carried out for {@link ProcessCategoryType.CUTOFF} only as the re-pricing category is not started during the critical batch period (Cutoff to BOD). @param branchGroupCategory
IOpsPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IOpsPricingAdapter	validatePricingPollerStatus	Validates Pricing Poller status for EOD start @param processDate @param nextProcessDate @param branchGroupCode @throws FatalException	@see com.ofss.fc.app.adapter.ops.IOpsPricingAdapter#validatePricingPollerStatus(com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date) @author abhishekbh @since R2.2
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	assertFraudPolicies	This method will invoke the OAAM policies to check for Fraud Transactions and take necessary action defined in policy. It returns List of Actions which is a result of OAAM policy execution. @param,MI,HashMap,serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap. @return String - Action returned by OAAM - BLOCK, Challenge1FA, Challenge2FA, Challenge1FADelay, Challenge2FADelay @throws FatalException	This method will invoke the polices and rules set in OAAM. @param HashMap - serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	assertFraudPolicies	This method will invoke the OAAM policies to check for Fraud Transactions and take necessary action defined in policy. It returns List of Actions which is a result of OAAM policy execution. @param,MI,HashMap,serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap. @return String - Action returned by OAAM - BLOCK, Challenge1FA, Challenge2FA, Challenge1FADelay, Challenge2FADelay @throws FatalException	This method will invoke the OAAM policies to check for Fraud Transactions and take necessary action defined in policy. It returns List of Actions which is a result of OAAM policy execution. @param,MI,HashMap,serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap. @return String - Action returned by OAAM - BLOCK, Challenge1FA, Challenge2FA, Challenge1FADelay, Challenge2FADelay @throws FatalException
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	assertFraudPolicies	This method will invoke the OAAM policies to check for Fraud Transactions and take necessary action defined in policy. It returns List of Actions which is a result of OAAM policy execution. @param,MI,HashMap,serviceInputValues, ,This parameter captures the method arguments.  The method parameters  and arguments are stored in key-value  pair the HashMap. @return String - Action returned by OAAM - BLOCK, Challenge1FA, Challenge2FA, Challenge1FADelay, Challenge2FADelay @throws FatalException	Method which returns FraudAssertionAdapter or mock adaptor @param String adapter @param NameValuePair [] nameValues @return Object

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	addUnApprovedTransactionRequest	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException	This method will save fraudulent unapproved request. If authentication is required then fraud request will be created or if authentication is completed (with or without delay) fraud request will be updated.
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	addUnApprovedTransactionRequest	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	addUnApprovedTransactionRequest	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	updateUnApprovedTransactionRequest	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException	This method will save fraudulent unapproved request. If authentication is required then fraud request will be created or if authentication is completed (with or without delay) fraud request will be updated.
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	updateOAMTransactionStatus	This method will update OAM transaction	This operation is used for updating OAM transaction status to the input status. @param workItem, ApprovalWorkItem, Workitem against which OAM transaction status is to be updated. @param transactionStatus, FraudAssertionTransactionStatus, Status to be set for OAM transaction. @return
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	updateOAMTransactionStatus	This method will update OAM transaction	This method marks OAM Transaction Status as Reject/ Pending and updates same using OAM APIs.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	updateOAMTransactionStatus	This method will update OAAM transaction	This operation is used for updating OAAM transaction status to the input status. @param workItem, ApprovalWorkItem, Workitem against which OAAM transaction status is to be updated. @param transactionStatus, FraudAssertionTransactionStatus, Status to be set for OAAM transaction. @return
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	markOAMTransactionAsFailed	This method will set OBP and OAAM transaction status on thread as failed.	This method will mark OAAM Transaction status on rollback to Failed/ Pending Mark status will be Failed - If the transaction is failed because of some business error other than approval required. Mark status will be Pending - If the transaction requires approval. @param isTransactionFailure @param isApprovalRequired
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	markOAMTransactionAsPendingForApprovals	This method will set OAAM transaction status as Pending if it was already set to Success. This indicates Approval required status.	This method will mark OAAM Transaction status on rollback to Failed/ Pending Mark status will be Failed - If the transaction is failed because of some business error other than approval required. Mark status will be Pending - If the transaction requires approval. @param isTransactionFailure @param isApprovalRequired
IFraudAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionAdapter	isDelayFraudTransaction	This method returns true if fraudulent transaction is to be sent for approval.	This method will trigger the accounting event. The accounting event will be transaction nature specific and hence the subclass will have to provide the correct event code and its associated values Call the accounting adapter and raise the accounting event. Before raising the event, the system will also create the array of {@link AccountingEventAmountTagDetailsDTO} based on the account debit response. <style type="text/css">table.tableizer-table {border: 1px solid #CCC; font-family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} </style>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer- firstrow"&gt; &lt;th&gt;Actual Event&lt;/th&gt; &lt;th&gt;Event Code&lt;/th&gt; &lt;th&gt;Dr Transaction Code&lt;/th&gt; &lt;th&gt;Netting Indicator&lt;/th&gt; &lt;th&gt;Account Role Code - Debit&lt;/th&gt; &lt;th&gt;Amount Tag - Debit&lt;/th&gt; &lt;th&gt;Cr Transaction Code&lt;/th&gt; &lt;th&gt;Account Role Code - Credit&lt;/th&gt; &lt;th&gt;&amp;nbsp;&lt;/th&gt; &lt;th&gt;Module responsibility&lt;/th&gt; &lt;th&gt;GAAP Code&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PRECUTOFF&lt;/td&gt; &lt;td&gt;C54&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E58&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_CASA_PSTCUTOFF&lt;/td&gt; &lt;td&gt;C55&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E59&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment from GL-before cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PRECUTOFF&lt;/td&gt; &lt;td&gt;E60&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;E61&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Outward Payment from GL-after cutoff&lt;/td&gt; &lt;td&gt;PY_OUTPAY_DEBIT_GL_PSTCUTOFF&lt;/td&gt; &lt;td&gt;E62&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E63&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_MOVEMENT_HELDOVR_ENDPOINT&lt;/td&gt; &lt;td&gt;E64&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E65&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-before cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_PRECUT&lt;/td&gt; &lt;td&gt;E66&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/ td&gt; &lt;td&gt;C56&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from CASA-after cutoff&lt;/td&gt; &lt;td&gt;PY_RECALL_OUTPAY_CASA_POSTCUT&lt;/td&gt; &lt;td&gt;E67&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;C57&lt;/td&gt; &lt;td&gt;PAYMENTS_CASA_ACCOUNT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Outward Payment from GL&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;PY_RECALL_OUTPAY_GL&lt;/td&gt; &lt;td&gt;E68&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT_EQVLNT&lt;/td&gt; &lt;td&gt;E69&lt;/td&gt; &lt;td&gt;PAYMENTS_RESPECTIVE_GL&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Recall-Heldover to endpoint movement&lt;/td&gt; &lt;td&gt;PY_RECALL_MVMT_HELDOVR_ENDPT&lt;/td&gt; &lt;td&gt;E70&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E71&lt;/td&gt; &lt;td&gt;HELDOVER&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Outward Payment for ASP Verification&lt;/td&gt; &lt;td&gt;PY_OUTPAY_ASP_VERIFY&lt;/td&gt; &lt;td&gt;E75&lt;/td&gt; &lt;td&gt;N&lt;/td&gt; &lt;td&gt;PAYMENTS_MISC_EXP&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;E76&lt;/td&gt; &lt;td&gt;PAYMENTS_END_POINT&lt;/td&gt; &lt;td&gt;OUTWARD_PAYMENT_AMOUNT&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;AL&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param transactionLog @param amountTagDetailsDTO @param messageLogItem @throws FatalException </pre>
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	executeRule	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId @param domainId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param domainId @return @throws FatalException	
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	executeRule	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId @param domainId @return @throws FatalException	This method executes the policy, whether its Inbound or outbound. It return DocumentPolicyExecutionResponseDTO which consist of DocumentCategoryDTO[] and a flag which will denote whether the input service event Id is same as that of the service id fetched from the Policy. @param sessionContext @param serviceId @param domainId @param domainCategoryId @param policyLinkageDTO @param factDTO @return @throws FatalException
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	executeRule	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId @param domainId @return @throws FatalException	This represents the addDocumentPolicy method of the entity. In addDocumentPolicy() a Policy is added for selected Domain and Domain Category. Depending on the Entity Linkage level selected,number of services are added,and under each Service....list of Document Categories are added. Rules are published for a service. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentPolicyDTO,documentPolicyDTO, ,DocumentPolicyDTO @fcb.param MI,RuleContext,ruleContext, ,RuleContext object containing the details of the rules related to the policy @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	executeRule	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId @param domainId @return @throws FatalException	This method executes a rule using the RuleExecutionApplicationService to find out the template id to be applied for the report. @param sessionContext @param ruleId @param domainCategoryId @param domainId @return @throws FatalException
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	executeRule	This method executes a rule using the RuleExecutionApplicationService. It returns a boolean which indicates whether the documentCategories must be returned for the service or not. @param sessionContext @param serviceId @param ruleId @param domainCategoryId @param domainId @return @throws FatalException	This method executes a rule using the RuleExecutionApplicationService. @param sessionContext @param ruleId @param domainCategoryId @param domainId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	publishRule	<p>This method will publish the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;Generate the code for this Rule&lt;/li&gt; &lt;li&gt;Persist the Rule&lt;/li&gt; &lt;li&gt;Save Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleDefinition Definition of Rule. Can be DecisionTable or GuidedRule. @return TransactionStatus Contains status of the transaction. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>	<p>R This method will publish the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;Generate the code for this Rule&lt;/li&gt; &lt;li&gt;Persist the Rule&lt;/li&gt; &lt;li&gt;Save Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleDefinition Definition of Rule. Can be DecisionTable or GuidedRule. @return TransactionStatus Contains status of the transaction. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	publishRule	<p>This method will publish the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;Generate the code for this Rule&lt;/li&gt; &lt;li&gt;Persist the Rule&lt;/li&gt; &lt;li&gt;Save Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleDefinition Definition of Rule. Can be DecisionTable or GuidedRule. @return TransactionStatus Contains status of the transaction. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>	<p>This represents the addDocumentPolicy method of the entity. In addDocumentPolicy() a Policy is added for selected Domain and Domain Category. Depending on the Entity Linkage level selected,number of services are added,and under each Service....list of Document Categories are added. Rules are published for a service. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,DocumentPolicyDTO,documentPolicyDTO, ,DocumentPolicyDTO @fcb.param MI,RuleContext,ruleContext, ,RuleContext object containing the details of the rules related to the policy @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	publishRule	<p>This method will publish the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;Generate the code for this Rule&lt;/li&gt; &lt;li&gt;Persist the Rule&lt;/li&gt; &lt;li&gt;Save Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleDefinition Definition of Rule. Can be DecisionTable or GuidedRule. @return TransactionStatus Contains status of the transaction. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>	<p>This represents the updateDocumentPolicy method of the entity. In updateDocumentPolicy, initially the Document Policy is fetched by Id, if it exists than the attributes of the Document Policy is updated. For example, a new Service can be added under the same Policy, and new categories can be linked same service. @fcb.param MI, com.ofss.fc.app.context.SessionContext, sessionContext, , This parameter captures the basic information about the user of the system. &lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI, DocumentPolicyDTO, documentPolicyDTO, documentPolicyDTO TODO: Explain the documentPolicyDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	publishRule	<p>This method will publish the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;Generate the code for this Rule&lt;/li&gt; &lt;li&gt;Persist the Rule&lt;/li&gt; &lt;li&gt;Save Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleDefinition Definition of Rule. Can be DecisionTable or GuidedRule. @return TransactionStatus Contains status of the transaction. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>	<p>This represents the addLayoutPolicy method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,LayoutPolicyDTO,layoutPolicyDTO,layoutPolicyDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	viewLatestRuleForProcessDate	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @param date Date with respect to which Rule Definition object needs to be fetched @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	viewLatestRuleForProcessDate	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>	This represents the updateDocumentPolicy method of the entity. In updateDocumentPolicy, initially the Document Policy is fetched by Id, if it exists than the attributes of the Document Policy is updated. For example, a new Service can be added under the same Policy, and new categories can be linked same service. @fcb.param MI, com.ofss.fc.app.context.SessionContext, sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI, DocumentPolicyDTO, documentPolicyDTO, documentPolicyDTO TODO: Explain the documentPolicyDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	viewLatestRuleForProcessDate	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>	This represents the fetchDocumentPolicy method of the entity. In this method already added Document Policy is fetched using the Document Policy Id,if the Policy is not present that method will throw RECORD NOT FOUND exception. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,DocumentPolicyDTO,documentPolicyDTO, ,documentPolicyDTO TODO: Explain the documentPolicyDTO above @return {@link com.ofss.fc.app.dto.content.DocumentPolicyInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	viewLatestRuleForProcessDate	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>	This represents the updateLayoutPolicy method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,LayoutPolicyDTO,layoutPolicyDTO, ,layoutPolicyDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	viewLatestRuleForProcessDate	This method will return latest Rule Definition object corresponding to given ruleId, with respect to process date @param sessionContext Information about context in which Rule is used. @param ruleId Id of the Rule @return RuleDefinitionInquiryResponse Containing information about details of decision table @throws FatalException <ul> Error Reason may be one of the following <li>I/O error occurs while deserializing Rule object</li></ul>	This represents the fetchLayoutPolicy method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,LayoutPolicyDTO,layoutPolicyDTO,layoutPolicyDTO @return {@link com.ofss.fc.app.dto.content.outbound.LayoutPolicyInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	deleteDraft	This method will delete rule draft corresponding to given ruleId and effectiveDate. @param sessionContext Information about context in which Rule is used. @param ruleId id of Rule to be modified @param effectiveDate effective date of Rule to be modified @throws FatalException <ul> Error Reason may be one of the following <li> {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_CANT_MODIFY_DELETE_PUBLISHED_DRAFT}</li> </ul>	This method will delete rule draft corresponding to given ruleId and effectiveDate. @param sessionContext Information about context in which Rule is used. @param ruleId id of Rule to be modified @param effectiveDate effective date of Rule to be modified @throws FatalException <ul> Error Reason may be one of the following <li> {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_CANT_MODIFY_DELETE_PUBLISHED_DRAFT}</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	deleteDraft	<p>This method will delete rule draft corresponding to given ruleId and effectiveDate. @param sessionContext Information about context in which Rule is used. @param ruleId id of Rule to be modified @param effectiveDate effective date of Rule to be modified @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt; {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_CANT_MODIFY_DELETE_PUBLISHED_DRAFT}&lt;/li&gt; &lt;/ul&gt;</p>	<p>This represents the deleteDocumentPolicy method of the entity.This method is called to delete the Record.Before deleting the Record a call will be made to fetch Document Policy,if the Record is present than that record will be Deleted by calling deleteDocumentPolicy(). @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,DocumentPolicyDTO,documentPolicyDTO,documentPolicyDTO TODO: Explain the documentPolicyDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	modifyRule	<p>This method will modify the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;If Rule is not yet effective then it will override old rule definition. Otherwise will store this as new Rule. In order to store this as new Rule effective date must me different than old Rule&lt;/li&gt; &lt;li&gt;Generate the code for modified Rule&lt;/li&gt; &lt;li&gt;Persist the modified Rule&lt;/li&gt; &lt;li&gt;Save modified Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleId Id of Rule being modified @param effectiveDate Effective date of Rule being modified @param newRuleDefinition New Definition of Rule. Can be DecisionTable or GuidedRule. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_RULE_EFFECTIVE_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>	<p>This method will modify the Rule. Method will do following operations: &lt;ul&gt; &lt;li&gt;If Rule is not yet effective then it will override old rule definition. Otherwise will store this as new Rule. In order to store this as new Rule effective date must me different than old Rule&lt;/li&gt; &lt;li&gt;Generate the code for modified Rule&lt;/li&gt; &lt;li&gt;Persist the modified Rule&lt;/li&gt; &lt;li&gt;Save modified Rule in Draft with published flag as Y&lt;/li&gt; &lt;/ul&gt; @param sessionContext Information about context in which Rule is used. @param ruleId Id of Rule being modified @param effectiveDate Effective date of Rule being modified @param newRuleDefinition New Definition of Rule. Can be DecisionTable or GuidedRule. @throws FatalException &lt;ul&gt; Error Reason may be one of the following &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_RULE_EFFECTIVE_TODAY}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION}&lt;/li&gt; &lt;/ul&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	modifyRule	<p>This method will modify the Rule. Method will do following operations:</p> <ul style="list-style-type: none"> <li>If Rule is not yet effective then it will override old rule definition. Otherwise will store this as new Rule. In order to store this as new Rule effective date must be different than old Rule</li> <li>Generate the code for modified Rule</li> <li>Persist the modified Rule</li> <li>Save modified Rule in Draft with published flag as Y</li> </ul> <p>@param sessionContext Information about context in which Rule is used.  @param ruleId Id of Rule being modified  @param effectiveDate Effective date of Rule being modified  @param newRuleDefinition New Definition of Rule. Can be DecisionTable or GuidedRule.  @throws FatalException</p> <p>Error Reason may be one of the following</p> <ul style="list-style-type: none"> <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE</a></li> <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_RULE_EFFECTIVE_TODAY</a></li> <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION</a></li> </ul>	<p>This represents the updateDocumentPolicy method of the entity. In updateDocumentPolicy, initially the Document Policy is fetched by Id, if it exists then the attributes of the Document Policy is updated. For example, a new Service can be added under the same Policy, and new categories can be linked same service.</p> <p>@fcb.param MI, com.ofss.fc.app.context.SessionContext, sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p>@fcb.param MI, DocumentPolicyDTO, documentPolicyDTO, documentPolicyDTO TODO: Explain the documentPolicyDTO above  @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for <i>Transaction Failure</i> section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentRuleAdapter	modifyRule	<p>This method will modify the Rule. Method will do following operations: <ul style="list-style-type: none"> <li>If Rule is not yet effective then it will override old rule definition. Otherwise will store this as new Rule. In order to store this as new Rule effective date must be different than old Rule</li> <li>Generate the code for modified Rule</li> <li>Persist the modified Rule</li> <li>Save modified Rule in Draft with published flag as Y</li> </ul> @param sessionContext Information about context in which Rule is used. @param ruleId Id of Rule being modified @param effectiveDate Effective date of Rule being modified @param newRuleDefinition New Definition of Rule. Can be DecisionTable or GuidedRule. @throws FatalException <ul style="list-style-type: none"> Error Reason may be one of the following <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULE_EFFECTIVE_DATE</a></li> <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_RULE_EFFECTIVE_TODAY</a></li> <li><a href="#">com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_INVALID_RULE_DEFINITION</a></li> </ul> </p>	<p>This represents the updateLayoutPolicy method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,LayoutPolicyDTO,layoutPolicyDTO,layoutPolicyDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchLoanAdapter	fetchLoanAccountDetails	<p>&lt;p&gt;This method returns the LoanAccountNotificationReponse that consist of various things like total arrears amount, advance payment, next due date etc. @param accountNumber @return the &lt;tt&gt;LoanAccountNotificationReponse&lt;/tt&gt; object. @throws FatalException</p>	
IBranchLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchLoanAdapter	fetchLoanAccountDetails	<p>&lt;p&gt;This method returns the LoanAccountNotificationReponse that consist of various things like total arrears amount, advance payment, next due date etc. @param accountNumber @return the &lt;tt&gt;LoanAccountNotificationReponse&lt;/tt&gt; object. @throws FatalException</p>	<p>&lt;p&gt;This method will call the BranchLOANAdapter to fetch the basic account details for a Loan Account. it will return LoanAccountNotificationReponse that will contain details like due outgoing amount, total arrears amount as well as account status. &lt;/p&gt; @param accountNumber @return @throws FatalException</p>
IBranchLoan Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchLoanAdapter	fetchLoanAccountDetails	<p>&lt;p&gt;This method returns the LoanAccountNotificationReponse that consist of various things like total arrears amount, advance payment, next due date etc. @param accountNumber @return the &lt;tt&gt;LoanAccountNotificationReponse&lt;/tt&gt; object. @throws FatalException</p>	<p>This method will return LoanServeInquiryResponse for loanAccount .This is required for Loan Serve Make Payment Transaction. @param sessionContext @param accountId @return LoanServeInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTaxationAdapter	notifyIncomeForTaxation	<p>This method is called to record income for an arrangement.&lt;br&gt;Whenever an arrangement earns interest it is considered as an income and that fact needs to be recorded for the purpose of levying tax.&lt;br&gt;This method will internally delegate the call to recordIncome method of {@link TaxManagementService}, so that tax module registers this income in its income pool.&lt;br&gt; @param productClass {@link ProductClass} to which this arrangement belongs, for which income is being notified. @param arrangementNumber arrangement number, for which income is being notified. @param partyId Identifier of Party of this arrangement, for which income is being notified. @param incomeAmount Interest amount which is being notified as Income liable for tax. @param requestDate Date on which request is being made for income notification.</p>	<p>This method is called to record income for an arrangement.&lt;br&gt; Whenever an arrangement earns interest it is considered as an income and that fact needs to be recorded for the purpose of levying tax.&lt;br&gt;This method will internally delegate the call to recordIncome method of {@link TaxManagementService}, so that tax module registers this income in its income pool.&lt;br&gt; @param productClass {@link ProductClass} to which this arrangement belongs, for which income is being notified. @param arrangementNumber arrangement number, for which income is being notified. @param partyId Identifier of Party of this arrangement, for which income is being notified. @param incomeAmount Interest amount which is being notified as Income liable for tax. @param requestDate Date on which request is being made for income notification.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTaxationAdapter	notifyIncomeForTaxation	<p>This method is called to record income for an arrangement.&lt;br&gt;Whenever an arrangement earns interest it is considered as an income and that fact needs to be recorded for the purpose of levying tax.&lt;br&gt;This method will internally delegate the call to recordIncome method of {@link TaxManagementService}, so that tax module registers this income in its income pool.&lt;br&gt; @param productClass {@link ProductClass} to which this arrangement belongs, for which income is being notified. @param arrangementNumber arrangement number, for which income is being notified. @param partyId Identifier of Party of this arrangement, for which income is being notified. @param incomeAmount Interest amount which is being notified as Income liable for tax. @param requestDate Date on which request is being made for income notification.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTaxationAdapter	notifyIncomeForTaxation	<p>This method is called to record income for an arrangement.&lt;br&gt;Whenever an arrangement earns interest it is considered as an income and that fact needs to be recorded for the purpose of levying tax.&lt;br&gt;This method will internally delegate the call to recordIncome method of {@link TaxManagementService}, so that tax module registers this income in its income pool.&lt;br&gt; @param productClass {@link ProductClass} to which this arrangement belongs, for which income is being notified. @param arrangementNumber arrangement number, for which income is being notified. @param partyId Identifier of Party of this arrangement, for which income is being notified. @param incomeAmount Interest amount which is being notified as Income liable for tax. @param requestDate Date on which request is being made for income notification.</p>	<p>This method is called to record income for an arrangement.&lt;br&gt; Whenever an arrangement earns interest it is considered as an income and that fact needs to be recorded for the purpose of levying tax.&lt;br&gt;This method will internally delegate the call to recordIncome method of {@link TaxManagementService}, so that tax module registers this income in its income pool.&lt;br&gt; @param productClass {@link ProductClass} to which this arrangement belongs, for which income is being notified. @param arrangementNumber arrangement number, for which income is being notified. @param partyId Identifier of Party of this arrangement, for which income is being notified. @param incomeAmount Interest amount which is being notified as Income liable for tax. @param requestDate Date on which request is being made for income notification.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	performTaxProcessing	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of <a href="#">TaxManagementService</a>.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside <a href="#">TaxProcessingResult</a> and return it to the caller.&lt;br&gt;Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new <a href="#">DepositInformation</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">TaxComputationRequest</a> object.</p>	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of <a href="#">TaxManagementService</a>.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside <a href="#">TaxProcessingResult</a> and return it to the caller.&lt;br&gt; Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new <a href="#">DepositInformation</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">AccountInformation</a> created in Step 2 in <a href="#">TaxComputationRequest#accountList</a>.&lt;/li&gt; &lt;li&gt;If (isBackDatedProcessing == true) then &lt;ol&gt; &lt;li&gt;Call <a href="#">TaxManagementService#processBackDatedTransaction()</a> method. Pass <a href="#">TaxComputationRequest</a> object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;Call <a href="#">TaxManagementService#processBackDatedTransaction()</a> method. Pass <a href="#">TaxComputationRequest</a> object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Above method will return <a href="#">TaxComputationResponse</a> object.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxProcessingResult</a> object populate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountInformation} created in Step 2 in {<a href="#">@link TaxComputationRequest#accountList</a>}.  If (isBackDatedProcessing == true) then  <ol style="list-style-type: none"> <li>Call {<a href="#">@link TaxManagementService#processBackDatedTransaction()</a>} method. Pass {<a href="#">@link TaxComputationRequest</a>} object created in Step 3 as input to it.</li> </ol> <li>Else  <ol style="list-style-type: none"> <li>Call {<a href="#">@link TaxManagementService#processBackDatedTransaction()</a>} method. Pass {<a href="#">@link TaxComputationRequest</a>} object created in Step 3 as input to it.</li> </ol> </li> <p>Above method will return {<a href="#">@link TaxComputationResponse</a>} object.  Create new {<a href="#">@link TaxProcessingResult</a>} object populate information in it using {<a href="#">@link TaxComputationResponse</a>} object obtained in Step 6.  Return {<a href="#">@link TaxProcessingResult</a>} object to caller.  @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {<a href="#">@link TaxComputationResult</a>} object containing details of tax computed along with pre and post tax Interest amounts.</p> </p>	<p>information in it using {<a href="#">@link TaxComputationResponse</a>} object obtained in Step 6.  Return {<a href="#">@link TaxProcessingResult</a>} object to caller.  @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {<a href="#">@link TaxComputationResult</a>} object containing details of tax computed along with pre and post tax Interest amounts.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	performTaxProcessing	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of {@link TaxManagementService}.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside {@link TaxProcessingResult} and return it to the caller.&lt;br&gt;Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new {@link DepositInformation} object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new {@link AccountInformation} object.&lt;br&gt; Populate {@link ProductClass} in it. Also put {@link DepositInformation} object object created in Step 1 in {@link AccountInformation#depositList}.&lt;/li&gt; &lt;li&gt;Create new {@link TaxComputationRequest} object.&lt;br&gt; Populate partyId in it. Also put {@link</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountInformation} created in Step 2 in {@link TaxComputationRequest#accountList}.</p> <p>If (isBackDatedProcessing == true) then</p> <ol style="list-style-type: none"> <li>Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.</li> </ol> <p>Else</p> <ol style="list-style-type: none"> <li>Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.</li> </ol> <p>Above method will return {@link TaxComputationResponse} object.</p> <p>Create new {@link TaxProcessingResult} object populate information in it using {@link TaxComputationResponse} object obtained in Step 6.</p> <p>Return {@link TaxProcessingResult} object to caller.</p> <p>@param taxProcessingRequest Arrangement number for which Tax processing is being done. return {@link TaxComputationResult} object containing details of tax computed along with pre and post tax Interest amounts.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	performTaxProcessing	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of <a href="#">{@link TaxManagementService}</a>.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside <a href="#">{@link TaxProcessingResult}</a> and return it to the caller.&lt;br&gt;Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new <a href="#">{@link DepositInformation}</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">{@link AccountInformation}</a> object.&lt;br&gt; Populate <a href="#">{@link ProductClass}</a> in it. Also put <a href="#">{@link DepositInformation}</a> object object created in Step 1 in <a href="#">{@link AccountInformation#depositList}</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">{@link TaxComputationRequest}</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">{@link</a></p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountInformation} created in Step 2 in {@link TaxComputationRequest#accountList}.</p> <p>If (isBackDatedProcessing == true) then</p> <ol style="list-style-type: none"> <li>Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.</li> </ol> <p>Else</p> <ol style="list-style-type: none"> <li>Call {@link TaxManagementService#processBackDatedTransaction()} method. Pass {@link TaxComputationRequest} object created in Step 3 as input to it.</li> </ol> <p>Above method will return {@link TaxComputationResponse} object.</p> <p>Create new {@link TaxProcessingResult} object populate information in it using {@link TaxComputationResponse} object obtained in Step 6.</p> <p>Return {@link TaxProcessingResult} object to caller.</p> <p>@param taxProcessingRequest Arrangement number for which Tax processing is being done. return {@link TaxComputationResult} object containing details of tax computed along with pre and post tax Interest amounts.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	performTaxProcessing	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of <a href="#">TaxManagementService</a>.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside <a href="#">TaxProcessingResult</a> and return it to the caller.&lt;br&gt;Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new <a href="#">DepositInformation</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">TaxComputationRequest</a> object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;</p>	<p>This method is used to perform tax processing for income that is already registered with Taxation module.&lt;br&gt;This method will internally delegate tax processing call to computeAndReturnTax method of <a href="#">TaxManagementService</a>.&lt;br&gt; For given arrangement number Taxation module will perform tax computation and it will pass Accounting entries related to tax.&lt;br&gt; Tax module will return Interest Income before tax, Tax amount and Rounded Interest Income after tax.&lt;br&gt;This method will wrap above details inside <a href="#">TaxProcessingResult</a> and return it to the caller.&lt;br&gt; Following is the algorithm for this method:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Create new <a href="#">DepositInformation</a> object.&lt;br&gt; Populate eventId, arrangementNumber, valueDate in it.&lt;/li&gt; &lt;li&gt;Create new <a href="#">AccountInformation</a> object.&lt;br&gt; Populate <a href="#">ProductClass</a> in it. Also put <a href="#">DepositInformation</a> object object created in Step 1 in <a href="#">AccountInformation#depositList</a>.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxComputationRequest</a> object.&lt;br&gt; Populate partyId in it. Also put <a href="#">AccountInformation</a> created in Step 2 in <a href="#">TaxComputationRequest#accountList</a>.&lt;/li&gt; &lt;li&gt;If (isBackDatedProcessing == true) then &lt;ol&gt; &lt;li&gt;Call <a href="#">TaxManagementService#processBackDatedTransaction()</a> method. Pass <a href="#">TaxComputationRequest</a> object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Else &lt;ol&gt; &lt;li&gt;Call <a href="#">TaxManagementService#processBackDatedTransaction()</a> method. Pass <a href="#">TaxComputationRequest</a> object created in Step 3 as input to it.&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Above method will return <a href="#">TaxComputationResponse</a> object.&lt;/li&gt; &lt;li&gt;Create new <a href="#">TaxProcessingResult</a> object populate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountInformation} created in Step 2 in {<a href="#">@link TaxComputationRequest#accountList</a>}.          If (isBackDatedProcessing == true) then          Call {<a href="#">@link TaxManagementService#processBackDatedTransaction()</a>} method. Pass {<a href="#">@link TaxComputationRequest</a>} object created in Step 3 as input to it.          Else          Call {<a href="#">@link TaxManagementService#processBackDatedTransaction()</a>} method. Pass {<a href="#">@link TaxComputationRequest</a>} object created in Step 3 as input to it.          Above method will return {<a href="#">@link TaxComputationResponse</a>} object.          Create new {<a href="#">@link TaxProcessingResult</a>} object populate information in it using {<a href="#">@link TaxComputationResponse</a>} object obtained in Step 6.          Return {<a href="#">@link TaxProcessingResult</a>} object to caller.          @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {<a href="#">@link TaxComputationResult</a>} object containing details of tax computed along with pre and post tax Interest amounts.</p>	<p>information in it using {<a href="#">@link TaxComputationResponse</a>} object obtained in Step 6.          Return {<a href="#">@link TaxProcessingResult</a>} object to caller.          @param taxProcessingRequest Arrangement number for which Tax processing is being done. return {<a href="#">@link TaxComputationResult</a>} object containing details of tax computed along with pre and post tax Interest amounts.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	getTaxationLevel	<p>There can be different levels at which tax can be computed and levied.&lt;br&gt; For example:&lt;br&gt; &lt;ul&gt;&lt;li&gt;Deposit Level(In case of Term Deposit): Tax computed and levied separately for each Term Deposit. Income generated due to that Term Deposit will only be taken into consideration while computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Account Level: Tax computed and levied separately for each Account. In case of TD income of all deposits under an Account will be considered during computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Party Level: Tax computed and levied separately for each Party. Income generated due to all accounts of party will be taken into account while computation and deduction of tax.&lt;/li&gt;&lt;/ul&gt; The above mentioned parameter is maintained with Taxation Module.&lt;br&gt;This method is used to fetch this parameter from Taxation Module.&lt;br&gt; return Value of Taxation Level Parameter fetched from Taxation module.</p>	<p>There can be different levels at which tax can be computed and levied.&lt;br&gt; For example:&lt;br&gt; &lt;ul&gt;&lt;li&gt;Deposit Level(In case of Term Deposit): Tax computed and levied separately for each Term Deposit. Income generated due to that Term Deposit will only be taken into consideration while computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Account Level: Tax computed and levied separately for each Account. In case of TD income of all deposits under an Account will be considered during computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Party Level: Tax computed and levied separately for each Party. Income generated due to all accounts of party will be taken into account while computation and deduction of tax.&lt;/li&gt;&lt;/ul&gt; The above mentioned parameter is maintained with Taxation Module.&lt;br&gt;This method is used to fetch this parameter from Taxation Module.&lt;br&gt; return Value of Taxation Level Parameter fetched from Taxation module.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	getTaxationLevel	<p>There can be different levels at which tax can be computed and levied.&lt;br&gt; For example:&lt;br&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li data-bbox="829 394 1283 618">&lt;li&gt;Deposit Level(In case of Term Deposit): Tax computed and levied separately for each Term Deposit. Income generated due to that Term Deposit will only be taken into consideration while computation and deduction of tax.&lt;/li&gt;</li> <li data-bbox="829 625 1283 813">&lt;li&gt;Account Level: Tax computed and levied separately for each Account. In case of TD income of all deposits under an Account will be considered during computation and deduction of tax.&lt;/li&gt;</li> <li data-bbox="829 820 1283 1008">&lt;li&gt;Party Level: Tax computed and levied separately for each Party. Income generated due to all accounts of party will be taken into account while computation and deduction of tax.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; The above mentioned parameter is maintained with Taxation Module.&lt;br&gt;This method is used to fetch this parameter from Taxation Module.&lt;br&gt; return Value of Taxation Level Parameter fetched from Taxation module.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	getTaxationLevel	<p>There can be different levels at which tax can be computed and levied.&lt;br&gt; For example:&lt;br&gt; &lt;ul&gt;&lt;li&gt;Deposit Level(In case of Term Deposit): Tax computed and levied separately for each Term Deposit. Income generated due to that Term Deposit will only be taken into consideration while computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Account Level: Tax computed and levied separately for each Account. In case of TD income of all deposits under an Account will be considered during computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Party Level: Tax computed and levied separately for each Party. Income generated due to all accounts of party will be taken into account while computation and deduction of tax.&lt;/li&gt;&lt;/ul&gt; The above mentioned parameter is maintained with Taxation Module.&lt;br&gt;This method is used to fetch this parameter from Taxation Module.&lt;br&gt; return Value of Taxation Level Parameter fetched from Taxation module.</p>	<p>There can be different levels at which tax can be computed and levied.&lt;br&gt; For example:&lt;br&gt; &lt;ul&gt;&lt;li&gt;Deposit Level(In case of Term Deposit): Tax computed and levied separately for each Term Deposit. Income generated due to that Term Deposit will only be taken into consideration while computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Account Level: Tax computed and levied separately for each Account. In case of TD income of all deposits under an Account will be considered during computation and deduction of tax.&lt;/li&gt;&lt;li&gt;Party Level: Tax computed and levied separately for each Party. Income generated due to all accounts of party will be taken into account while computation and deduction of tax.&lt;/li&gt;&lt;/ul&gt; The above mentioned parameter is maintained with Taxation Module.&lt;br&gt;This method is used to fetch this parameter from Taxation Module.&lt;br&gt; return Value of Taxation Level Parameter fetched from Taxation module.</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	simulateTax	This method is used to call Tax Module for simulating tax on input of adjustment amount. Based on the inputs TaxSimulationRequest is formed and a call to TaxManagementService is done which in turn returns the tax amount. @param accountId @param partyId @param adjustmentAmount @param productClass @throws FatalException	This method is used to call Tax Module for simulating tax on input of adjustment amount. Based on the inputs TaxSimulationRequest is formed and a call to TaxManagementService is done which in turn returns the tax amount. @param accountId @param partyId @param adjustmentAmount @param productClass @throws FatalException
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	simulateTax	This method is used to call Tax Module for simulating tax on input of adjustment amount. Based on the inputs TaxSimulationRequest is formed and a call to TaxManagementService is done which in turn returns the tax amount. @param accountId @param partyId @param adjustmentAmount @param productClass @throws FatalException	This method is used to call Tax Module for simulating tax on input of adjustment amount. Based on the inputs TaxSimulationRequest is formed and a call to TaxManagementService is done which in turn returns the tax amount. @param accountId @param partyId @param adjustmentAmount @param productClass @throws FatalException
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	getTaxForArrangementNumberforPeriod	This method fetches the total tax amount for an arrangement number for the input period. This method calls the {@link TaxationInquiryService#inquireTaxDetails(String, PeriodDTO, SessionContext)} method to fetch the tax amount. @param arrangementNumber @param startDate @param endDate @return	This method fetches the total tax amount for an arrangement number for the input period. This method calls the {@link TaxationInquiryService#inquireTaxDetails(String, PeriodDTO, SessionContext)} method to fetch the tax amount. @param arrangementNumber @param startDate @param endDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTaxationAdapter	getTaxForArrangementNumberforPeriod	This method fetches the total tax amount for an arrangement number for the input period. This method calls the {@link TaxationInquiryService#inquireTaxDetails(String, PeriodDTO, SessionContext)} method to fetch the tax amount. @param arrangementNumber @param startDate @param endDate @return	
IInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTaxationAdapter	getTaxForArrangementNumberforPeriod	This method fetches the total tax amount for an arrangement number for the input period. This method calls the {@link TaxationInquiryService#inquireTaxDetails(String, PeriodDTO, SessionContext)} method to fetch the tax amount. @param arrangementNumber @param startDate @param endDate @return	This method fetches the total tax amount for an arrangement number for the input period. This method calls the {@link TaxationInquiryService#inquireTaxDetails(String, PeriodDTO, SessionContext)} method to fetch the tax amount. @param arrangementNumber @param startDate @param endDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	taxAtInterestAdjustment	This method is called to deduct on the income specified in the input.  During Interest Adjustment when Tax applicable at Accrual and Accrued Interest is increased then in such scenario the tax   needs to be deducted on Adjusted income . In case the Accrued Interest is decreased then tax won't be refunded.  This method will called for the following cases :- <ol> <li>Tax at Accrual and Accrued Interest is increased. <li>Tax at Compounding and Comounded Interest is increased. <li>Tax at Capitalization and Capitalized Interest is increased. </ol> @param arrangementNumber @param partyId @param amount @param productClass @param dayCountConventionType @param daysForTaxCalculation @param eventID @param transactionReferenceNumber @param valueDate	This method is called to deduct on the income specified in the input.  During Interest Adjustment when Tax applicable at Accrual and Accrued Interest is increased then in such scenario the tax   needs to be deducted on Adjusted income . In case the Accrued Interest is decreased then tax won't be refunded.  This method will called for the following cases :- <ol> <li>Tax at Accrual and Accrued Interest is increased. <li>Tax at Compounding and Comounded Interest is increased. <li>Tax at Capitalization and Capitalized Interest is increased. </ol> @param arrangementNumber @param partyId @param amount @param productClass @param dayCountConventionType @param daysForTaxCalculation @param eventID @param transactionReferenceNumber @param valueDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	taxAtInterestAdjustment	This method is called to deduct on the income specified in the input.  During Interest Adjustment when Tax applicable at Accrual and Accrued Interest is increased then in such scenario the tax   needs to be deducted on Adjusted income . In case the Accrued Interest is decreased then tax won't be refunded.  This method will called for the following cases :- <ol> <li>Tax at Accrual and Accrued Interest is increased.<li>Tax at Compounding and Comounded Interest is increased.<li>Tax at Capitalization and Capitalized Interest is increased.</ol> @param arrangementNumber @param partyId @param amount @param productClass @param dayCountConventionType @param daysForTaxCalculation @param eventID @param transactionReferenceNumber @param valueDate	This method is called to deduct on the income specified in the input.  During Interest Adjustment when Tax applicable at Accrual and Accrued Interest is increased then in such scenario the tax   needs to be deducted on Adjusted income . In case the Accrued Interest is decreased then tax won't be refunded.  This method will called for the following cases :- <ol> <li>Tax at Accrual and Accrued Interest is increased. <li>Tax at Compounding and Comounded Interest is increased. <li>Tax at Capitalization and Capitalized Interest is increased. </ol> @param arrangementNumber @param partyId @param amount @param productClass @param dayCountConventionType @param daysForTaxCalculation @param eventID @param transactionReferenceNumber @param valueDate
InterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\interest\InterestTaxationAdapter	inquireTaxDetailsAfterRefund	This method will fetch the tax computed after refund for an arrangement for the date range specified in the input. @param arrangementNumber @param startDate @param endDate @return	This method will fetch the tax computed after refund for an arrangement for the date range specified in the input. @param arrangementNumber @param startDate @param endDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	inquireTaxDetailsAfterRefund	This method will fetch the tax computed after refund for an arrangement for the date range specified in the input. @param arrangementNumber @param startDate @param endDate @return	
IIInterestTaxationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IIInterestTaxationAdapter	inquireTaxDetailsAfterRefund	This method will fetch the tax computed after refund for an arrangement for the date range specified in the input. @param arrangementNumber @param startDate @param endDate @return	This method will fetch the tax computed after refund for an arrangement for the date range specified in the input. @param arrangementNumber @param startDate @param endDate @return
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getDaysPastDue		getDaysPastDue
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getCoverageRatio		
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getSacrificeAmount		
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getDaysToFirstInstallment		Calculates Days to First Installment i.e time from first scheduled installment date to posting date Modified for OBP22IUT-16324

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getNumberOfInstallments		getNumberOfInstallments
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getTotalPromiseDurationInDays		getTotalPromiseDurationInDays
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getPTPTType		getPTPTType
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getProductGroup		getProductGroup
IPromiseToPayWorkflowAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPromiseToPayWorkflowAdapter	getUserRole		getUserRole

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	fetchInstructions	This method fetches the Account Operating Instructions for a particular account. @param accountId Term deposit account id @return {@link OperatingInstructionInquiryResponse} Object containing information about instructions assigned to an account	This method fetches the Account Operating Instructions for a particular account. @param accountId Term deposit account id @return {@link OperatingInstructionInquiryResponse} object containing information about instructions assigned to an account @see com.ofss.fc.app.adapter.td.ITDAccountAdapter#fetchInstructions(java.lang.String)
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	fetchInstructions	This method fetches the Account Operating Instructions for a particular account. @param accountId Term deposit account id @return {@link OperatingInstructionInquiryResponse} Object containing information about instructions assigned to an account	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentId The account id from which this account created @return {@link TransactionStatus} status of the transaction which fetches account operation instructions and add in new account @see com.ofss.fc.app.adapter.td.ITDAccountAdapter#addInstructionForAccount(java.lang.String, java.lang.String)
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	fetchInstructions	This method fetches the Account Operating Instructions for a particular account. @param accountId Term deposit account id @return {@link OperatingInstructionInquiryResponse} Object containing information about instructions assigned to an account	This method will insert/update the account operating instructions for account when secondary applicant is added to the account from origination. @param sessionContext @param accountId @param holderPreferenceDTO @param operatingInstructionCategoryType @param individualPartyRole @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountAdapter	addInstructionForAccount	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentAccountId The account id from which this account created @return {@link TransactionStatus} Status of the transaction which fetches account operation instructions and add in new account	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentAccountId The account id from which this account created @return {@link TransactionStatus} status of the transaction which fetches account operation instructions and add in new account @see com.ofss.fc.app.adapter.td.ITDAccountAdapter#addInstructionForAccount(java.lang.String, java.lang.String)
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountAdapter	addInstructionForAccount	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentAccountId The account id from which this account created @return {@link TransactionStatus} Status of the transaction which fetches account operation instructions and add in new account	This method will insert/update the account operating instructions for account when secondary applicant is added to the account from origination. @param sessionContext @param accountId @param holderPreferenceDTO @param operatingInstructionCategoryType @param individualPartyRole @throws FatalException
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAccountAdapter	addInstructionForAccount	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentAccountId The account id from which this account created @return {@link TransactionStatus} Status of the transaction which fetches account operation instructions and add in new account	This method creates a New Account in the Product specified by PMU. It loads the existing Account and sets the Product Code to the Reinvest Product and carries forward the existing Account Attributes to the New Account. This method is called when Product level parameter is set to Reinvest at Maturity and Deposit level instructions have not been maintained. @param applicationContext @param reinvestProductCode @param deposit @param iTermDepositAccount @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	addInstructionForAccount	This method fetches the Account Operating Instructions from parent account and adds them in a new account. @param accountId Term deposit account id @param parentId The account id from which this account created @return {@link TransactionStatus} Status of the transaction which fetches account operation instructions and add in new account	This method creates a New Account in the Product specified by PMU. It loads the existing Account and sets the Product Code to the Reinvest Product and carries forward the existing Account Attributes to the New Account. This method is called when Product level parameter is set to Reinvest at Maturity and Deposit level instructions have not been maintained. @param applicationContext @param reinvestProductCode @param deposit @param iTermDepositAccount @return
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	updateAccountRepository	This method is called after successful account opening from origination. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link TransactionStatus} Status of the transaction which updates account repository after account opening from origination module has been done	This method is used to update the Account Configurations for a Term Deposit Account that has already been opened previously. It updates the following informations inside the Account Configuration.  1. Payout instructions provided for the account.  2. Financial and non financial Operating instructions for the account.  3. Default communication address for the account are updated.  @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,TDAccountConfigurationDTO, tdAccountConfigurationDTO, , This class represents the DTO class for term deposit account   configuration.This class contains information about the preferences given for the term deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	updateAccountRepository	This method is called after successful account opening from origination. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link TransactionStatus} Status of the transaction which updates account repository after account opening from origination module has been done	This method is called after the successful opening of an account. @param sessionContext @param tdAccountConfigurationDTO @throws FatalException
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	memoPresentOnAccount	The method checks if memo is present on a given account @param accountId Identifies a accountId for which the memo is to be checked. @return {@link MemoInquiryResponse} Signifies the memo information present on a TD account	The method checks if memo is present on a given account @param accountId Identifies a accountId for which the memo is to be checked. @return {@link MemoInquiryResponse} signifies the memo information present on a TD account @see com.ofss.fc.app.adapter.td.ITDAccountAdapter#memoPresentOnAccount(java.lang.String)
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	memoPresentOnAccount	The method checks if memo is present on a given account @param accountId Identifies a accountId for which the memo is to be checked. @return {@link MemoInquiryResponse} Signifies the memo information present on a TD account	The method checks if the memo is maintained for the TD account and if present,   the level of severity is logged for overridable business policy check.
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	deleteUnExecutedSystemDepositStatementRequest	This method deletes the pending System initiated statement requests @param accountId @param depositNumber @return	This method deletes the pending System initiated statement requests @param accountId @param depositNumber @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	deleteUnExecutedSystemDepositStatementRequest	This method deletes the pending System initiated statement requests @param accountId @param depositNumber @return	This method does the following :- <ul style="list-style-type: none"> <li>Deletes the pending statement request if any exist</li> <li>removes the already logged statement action</li> <li>logs new statementAction and also adds new statement requests</li> </ul> @param termDepositAccount
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	deleteUnExecutedSystemDepositStatementRequest	This method deletes the pending System initiated statement requests @param accountId @param depositNumber @return	This method does the following :- <ul style="list-style-type: none"> <li>Deletes the pending statement request if any exist</li> <li>removes the already logged statement action</li> <li>logs new statementAction and also adds new statement requests</li> </ul> @param termDepositAccount
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	orderTDStatementOnDeposit	As the statement requests are at account level , a new deposit will also need to be be part of statement generation. So we add a separate statement request for all new deposits on its opening. This method will loop through system initiated pending statement and add a separate request for the new deposit that has been opened. <p> It will return the statement request id of the newly added deposit request.If statement flag is not enabled, it will return null. </p> @param accountId @param depositNumber	As the statement requests are at account level , a new deposit will also need to be be part of statement generation. So we add a separate statement request for all new deposits on its opening. This method will loop through system initiated pending statement and add a separate request for the new deposit that has been opened. <p> It will return the statement request id of the newly added deposit request.If statement flag is not enabled, it will return null. </p> @param accountId @param depositNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	orderTDStatementOnDeposit	As the statement requests are at account level , a new deposit will also need to be be part of statement generation. So we add a separate statement request for all new deposits on its opening. This method will loop through system initiated pending statement and add a separate request for the new deposit that has been opened. <p>It will return the statement request id of the newly added deposit request.If statement flag is not enabled, it will return null. </p> @param accountId @param depositNumber	This method does the following :- <ul> <li>Deletes the pending statement request if any exist</li> <li>removes the already logged statement action</li> <li>logs new statementAction and also adds new statement requests</li> </ul> @param termDepositAccount
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	linkTDAccountToOtherBranch	This method is being used to call TD account transfer from Common Account Transfer. The operation linkTDAccountToOtherBranch is being called from AccountBranchTransferApplication Service API if module type for the account is TD. @param sessionContext @param AccountBranchTransferDetailsDTO @throws FatalException	Method linkModuleBranchTransfer is being used to decide the service call based on moduleType. ie if account type is CASA call method linkDDAccountToOtherBranch(sessionContext, ddAccountBranchTransferDetailsDTO) Method to link Module wise Account Branch Transfer.
ITDAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	isOfferAvailableForBundleReferenceId	This method checks whether offer is available for particular Bundle. @param offerId @param bundleReferenceId	This method checks whether offer is available for particular Bundle. @param offerId @param bundleReferenceId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAccountAdapter	isOfferAvailableForBundleReferenceId	This method checks whether offer is available for particular Bundle. @param offerId @param bundleReferenceId	Check for bundle details, set to null if the offer is not available for bundle reference id @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548
ICommonServiceDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceDDAAdapter	checkIfAccountsAreClosed		This is the helper method to check whether a particular BranchParameters record can be closed or not.  If the BranchParameters can be closed, it returns true, else it returns false. This method checks whether any account(s) is (are) opened under any domain category TD, CASA, Loan,  If no account is opened , the branch can be closed, otherwise, we should not allow to close the branch. @param sessionContext : {@link SessionContext} @param branchCode : String @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commonservice\ICommonServiceDDAAdapter	checkIfAccountsAreClosed		This method returns true, if all Demand Deposit accounts are closed under the given branch else returns false.
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchODpreferencesBasedOnProduct	This method fetched OD preferences for CASA product @param sessionContext @param productCode @return	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchODpreferencesBasedOnProduct	This method fetched OD preferences for CASA product @param sessionContext @param productCode @return	This method fetches the product preferences for the product code via the PMU adapter @param productCode @param accountType @return planCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchLoanpreferencesBasedOnProduct	This method fetched Loan preferences for LOAN product @param sessionContext @param productCode @return	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchLoanpreferencesBasedOnProduct	This method fetched Loan preferences for LOAN product @param sessionContext @param productCode @return	This method fetches the product preferences for the product code via the PMU adapter @param productCode @param accountType @return planCode
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchMinimumQualifyingAmount	This method fetches the Minimum Qualifying Amount required for classifying the asset. @param productCode @param currency @param businessUnit @return Money	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchMinimumQualifyingAmount	This method fetches the Minimum Qualifying Amount required for classifying the asset. @param productCode @param currency @param businessUnit @return Money	This method fetches the minimum qualifying amount for the product of the classification account @param productCode @param currency @param businessUnit @return minQualifyingAmount
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchMinimumQualifyingAmount	This method fetches the Minimum Qualifying Amount required for classifying the asset. @param productCode @param currency @param businessUnit @return Money	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchLoanPreferences	This method fetches the loan product preferences. @param productCode @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchLoanPreferences	This method fetches the loan product preferences. @param productCode @return	This method is called to perform the partial writeoff on the account. It first calls inquireAccountWriteOff() to fetch the arrear amounts due for the account. It then performs validation that the amount of arrears balances being written off are <= the due amounts. If this validation is successful it calls writeOffArrearPartially() of CasaArrearManagementService or LoanArrearManagementService via adapters for further processing, based on the account type. For Loan account it will check that if the capital writeoff amount is greater than the principal arrear raised and due and if process date is < maturity date of the account then it will reschedule the account based on default reschedule options. @param sessionContext @param accountWriteOffDTO @return transactionStatus
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchWriteoffRecoveryPreferences	This method fetches the product preferences related to recovery post write-off. @param productCode @param accountType @return	This method is to be called by the CASA/Loans module when there are payments for an Account which is either Written-off or Closed. The appropriation for the transaction amount will be done as per the written-off/closed appropriation sequence defined at the product level.   <b> The following events can occur: </b>  1. Validates against the Incoming Electronic Payments Allowed flag defined in the Product preference </b>  2. This method will throw a {@link BusinessException} if all the Written off amounts are already recovered</b>  3. If the transaction amount is lesser or equal to the unrecovered amount, the transaction amount will be appropriated for various arrear buckets as per the the written-off/closed appropriation sequence defined at the product level</b>  4. If the transaction amount is greater than the unrecovered amount, for the unrecovered amounts the appropriation is done to the respective arrear recovery ledgers. The excess

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				amounts which are received, will be parked in the Excess Payment Received Suspense GL and this additional money has to be refunded operationally  @param recoveryDTO @return {@link WriteoffRecoveryResponse} This class holds the AccountingEventAmountTagDetailsDTO array for which recovery is performed. @throws FatalException
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	fetchWriteoffRecoveryReferences	This method fetches the product preferences related to recovery post write-off. @param productCode @param accountType @return	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	getLedgerCodeForAccountRole	This method fetches the Ledger code for the accountrole and domainCategory @param domainCategory @param accountRole @return	
IAssetClassificationProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationProductAdapter	getLedgerCodeForAccountRole	This method fetches the Ledger code for the accountrole and domainCategory @param domainCategory @param accountRole @return	This method returns the Ledger code for Account Role: LN_INPAY_INTMED_GL @return
ITaxPMUApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxPMUApplicationAdapter	getInterestRules	@param productId @return	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxPMUApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxPMUApplicationAdapter	getInterestRules	@param productId @return	This method is called to fetch the {@link InterestRule} which is linked to the product.  For Interest calculation purpose there are certain parameters defined in {@link InterestRule} which are required .  The parameters which are required for Interest processing are as follows :-   <ul> <li> {@link DayCountConventionType} <li> {@link RoundingRule} </ul> @param productId @return InterestRule
ITaxPMUApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxPMUApplicationAdapter	getInterestRules	@param productId @return	
ITaxPMUApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxPMUApplicationAdapter	fetchProductGroup		This method fetches the product group of the input product code. @param productCode @return @throws FatalException
ITaxPMUApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxPMUApplicationAdapter	fetchProductGroup		
ILoanOffsetAccountApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanOffsetAccountApplicationAdapter	getAccountAverageBalance	This method will return average offset account balance between two dates, of the external CASA/TD Account linked to the loan account in Oracle application. @param applicationContext @param accountId @return BigDecimal @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAgentAdapter	linkAgentDetailsToAccount	This method calls AgentArrangement linkage repository if account has been sourced by an agent, agent will get linked to an td account. @param agentArrangementLinkageDTO {@link AgentArrangementLinkageDTO} Object containing agent details	
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAgentAdapter	linkAgentDetailsToAccount	This method calls AgentArrangement linkage repository if account has been sourced by an agent, agent will get linked to an td account. @param agentArrangementLinkageDTO {@link AgentArrangementLinkageDTO} Object containing agent details	This method is used to open an account. Algo  1. Get the party KYC status. If it is confirmed then mark the account status as confirm  else mark the account as AccountStatus.KYCCunconfirmed  2. Get the generated Account Number from account generation method  3. Call the create method of maintenanceDomainService to persist  4. Log account closure action  5. If account has been sourced by broker/introducer, then link agent details to an account 
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDAgentAdapter	fetchLatestSerialNumber	This method calls AgentArrangement linkage repository to fetch latest serial number for given arrangement id @param String arrangementId account id if it has been called at account opening or account id and deposit no separated by @return int serial number of provided arrangement id	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAgentAdapter	fetchLatestSerialNumber	This method calls AgentArrangement linkage repository to fetch latest serial number for given arrangement id @param String arrangementId account id if it has been called at account opening or account id and deposit no separated by @return int serial number of provided arrangement id	
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAgentAdapter	linkAgentDetailsToTermDeposit	This method calls AgentArrangement linkage repository if account has been sourced bt an agent, agent will get linked to the term deposit. @param agentArrangementLinkageDTO {@link AgentArrangementLinkageDTO} Object containing agent and commission details	
ITDAgentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDAgentAdapter	linkAgentDetailsToTermDeposit	This method calls AgentArrangement linkage repository if account has been sourced bt an agent, agent will get linked to the term deposit. @param agentArrangementLinkageDTO {@link AgentArrangementLinkageDTO} Object containing agent and commission details	This method will call TDAgentAdapter's linkAgentDetail to term deposit and will stamp all the commission details at term deposit level. @param agentArrangementLinkageDTO {@link AgentArrangementLinkageDTO} Object containing agent and commission details. @param payinResponse {@link PayinResponse} Object containing deposit details
IUisvcAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcAccountAdapter	fetchAccountByAccount		This method is used to log notifications for a party/account for the first time and then fetch it. @param sessionContext @param partyId @param accountId @param logNotificationService @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IUisvcAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcAccountAdapter	fetchAccountByAccount		<p>This method is used to fetch the account type of given account. whether it is CASA,DDA or LOAN.</p> @param sessionContext @param accountNumber @return @throws FatalException
IUisvcAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcAccountAdapter	fetchAccountByAccount		
IUisvcAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcAccountAdapter	fetchAccountByAccount		This method fetches notification logs. @param sessionContext @param partyId @param accountId @return @throws FatalException
IUisvcAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcAccountAdapter	fetchAccountByAccount		This method is used to close Deposit. @param accountId Account id to which deposit belongs @param depositId Identifier of Deposit to be closed @return {@link TransactionStatus} containing result of close operation @see com.ofss.fc.app.td.service.deposit.IDepositApplicationService#closeDeposit(com.ofss.fc.app.context.SessionContext, java.lang.String, int)
IAgentLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentLCMAdapter	fetchFacilityDetails	@param sessionContext @param facilityId @return @throws FatalException	This service will return AccountSummaryDTO for all the account under facilityId provided in input. @param facilityId :facility id of which all child Account will be returned back @return AccountSummaryResponse
IAgentLCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentLCMAdapter	fetchUltimateFacility	This will get the ultimate facility information of the given current facility ID @param sessionContext @param creditFacilityId @return @throws FatalException	This service will fetch the ultimate Parent Facility Id of the given creditFacility ID

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAgentLCM Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\agent\IAgentLCMAdapter	fetchUltimateFacility	This will get the ultimate facility information of the given current facility ID @param sessionContext @param creditFacilityId @return @throws FatalException	This will return ultimate parentFacilityId of the given Facility ID @param sessionContext @param facilityID @return
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	<ul style="list-style-type: none"> <li>The method is responsible for registering the account in the tax module during account opening.</li> <li>It calls {@link TaxManagementService#registerWithTax(depositInformation: DepositDetails)}</li> <li>DepositDetails required in input is populated as follows: <ul style="list-style-type: none"> <li>DepositDetails.partyId = partyId</li> <li>DepositDetails.AccountDepositDetails.accountId = accountId</li> <li>DepositDetails.AccountDepositDetails.ProductClass = DemandDeposit (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.branchCode = accountHomeBranch</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.arrangementNumber = accountId</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.referenceNumber = null</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.ArrangementStatus = Active (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.parentArrangementNumber = null</li> </ul> </li> </ul> @param partyId @param accountId @param accountHomeBranch @param productId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	<ul style="list-style-type: none"> <li>The method is responsible for registering the account with value date which is equivalent to account opening date in the tax module during account opening. It calls {@link TaxManagementService#registerWithTax(depositInformation: DepositDetails)}</li> <li>DepositDetails required in input is populated as follows: <ul style="list-style-type: none"> <li>DepositDetails.partyId = partyId</li> <li>DepositDetails.AccountDepositDetails.accountId = accountId</li> <li>DepositDetails.AccountDepositDetails.ProductClass = DemandDeposit (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.branchCode = accountHomeBranch</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.arrangementNumber = accountId</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.referenceNumber = null</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.ArrangementStatus = Active (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.parentArrangementNumber = null</li> </ul> </li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	<ul style="list-style-type: none"> <li>The method is responsible for registering the account in the tax module during account opening.</li> <li>It calls {@link TaxManagementService#registerWithTax(depositInformation: DepositDetails)}</li> <li>DepositDetails required in input is populated as follows: <ul style="list-style-type: none"> <li>DepositDetails.partyId = partyId</li> <li>DepositDetails.AccountDepositDetails.accountId = accountId</li> <li>DepositDetails.AccountDepositDetails.ProductClass = DemandDeposit (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.branchCode = accountHomeBranch</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.arrangementNumber = accountId</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.referenceNumber = null</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.ArrangementStatus = Active (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.parentArrangementNumber = null</li> </ul> </li> </ul>
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	Method intimateTaxation will be used to intimate the taxation @param partyId @param accountId @param sourceBranch @param destinationBranch @param productId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	<pre> This method contains the business logic for opening a demand deposit account based on the parameters {@link DemandDepositAccountOpeningDTO} passed by the calling module[eg Origination].&lt;br&gt; &lt;ol&gt; &lt;li&gt;Perform all validations using {@link DemandDepositAccountBusinessPolicy#validate()}. &lt;ul&gt; Following validations will be performed &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateProduct()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateProductExpiry()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validatePermissibleNoOfAccounts()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validatePartyId()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validatePartyRole()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateBranch()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateCurrency()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validatePartyBlacklisting()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateAccountRelationship()}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountBusinessPolicy#validateRestrictions()}&lt;/li&gt; &lt;/ul&gt; Generate the account number by using {@link DDAccountOpeningService#generateAccountNumber()}.&lt;/li&gt; &lt;li&gt;Call {@link this.deriveMinorAccountStatus()} to derive the minor status of the primary party and set </pre>





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>netRateDerived : The information object carrying the details of the rate negotiated. See <a href="#">{@link NetRateDerived}</a> for more information.</p> <p><a href="#">{@param partyToAccountRelationship[]}</a>: The information object carrying the party to account relationship.. See <a href="#">{@link PartyToAccountRelationship}</a> for more information.</p> <p><a href="#">{@ol}</a> For Account Opening:Product level banking facilities cannot be overridden(If at product debit card flag is false, <a href="#">{&lt;br&gt;}</a> at account level it cannot be set as true. If channelFacilitesDTO is null--&gt;banking facilities from product will be applicable. Maximum number of accounts that a party can have under the product as the primary (first) holder will be defined at the Product level. System should not permit CASA account opening if this account is breaching the set limit. System should permit CASA account opening only if the product code is active and not expired. It works as follows: <a href="#">{&lt;li&gt;}</a>Get the count of account for which the party is primary account holder by calling <code>IDDAPartyAdapter#fetchCountPrimaryAccountHolder(partyId : String)</code> <a href="#">{&lt;li&gt;}</a> Compare the count with the allowable count at the product level.<a href="#">{&lt;li&gt;}</a> Reject account opening if the product level count is less than or equal to the count for which party is primary account holder. <a href="#">{&lt;/pre&gt;}</a> <a href="#">{@param demandDepositAccount}</a> <a href="#">{@param interestTicketNumber}</a> <a href="#">{@param partyToAccountRelationship}</a> <a href="#">{@param isDocumentIncomplete}</a> <a href="#">{@param isCheckingAccount}</a> <a href="#">{@throws FatalException}</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	This method is used to perform different types of registration i.e registration with pricing, interest, NPA and taxation module @param demandDepositClonedAccount @param postingDate @param interestTicketNumber @param parentAccountId @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxation} It accepts following three inputs partyId, accountId, and accountBranch. @param partyId @param accountId @param accountBranch @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxationBackdated} It accepts following three inputs partyId, accountId, accountBranch, productCode and valueDate which is backdated account opening date. @param partyId @param accountId @param accountBranch @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	This method contains the business logic for opening a backdated demand deposit account based on the parameters {@link DemandDepositAccountOpeningDTO} passed by the calling module[eg Origination]. It contains the account opening date which is less than the current process date,it also contains the details of broker/introducer's which was used to open the backdated account. <ol>The following is the algorithm for the method:<li>Fetch all the product level parameters for given offer id {@link OfferInquiryApplicationService#fetchCASAOffer()}</li><li>Perform all validations using {@link DemandDepositAccountBusinessPolicy#validate()}.</li></ol><ul>Following validations will be performed in addition to the existing validations of account opening<li>{@link DemandDepositAccountBusinessPolicy#validateBatchRunning()}</li><li>{@link DemandDepositAccountBusinessPolicy#validatePeriodForBackDatedAccount}</li><li>{@link DemandDepositAccountBusinessPolicy#validateOfferForBackDated}</li></ul> <li>Generate the account number by using {@link DDAccountOpeningService#generateAccountNumber()}.</li> <li>Call {@link this.deriveMinorAccountStatus()} to derive the minor status of the primary party and set demandDepositAccount.AccountPreferences.minorAccountStatus = minorAccountStatus</li> <li>AccountPreferences is a component in the DemandDepositAccount</li> <li>Call {@link this.setAccountStatus()} to initialize the SystemDefined and UserDefined account statuses</li> <li>Call private method {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>setAgingStatusDates()} for initializing the aging statuses and dates.&lt;/li&gt; &lt;li&gt; Call {@link setDefaultAccountFacilities()} and {@link setDefaultAccountFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt; Call {@link setDefaultBankingFacilities()} and {@link setDefaultBankingFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt;Persist the account object to the database by calling {@link DemandDepositAccountRepository#insertDemandDepositAccount()}&lt;/li&gt; &lt;li&gt; Call private method {@link this.createPricingPlanAndEndDialogueForBackdate()} to create pricing plan in the pricing module using the back dated account opening date&lt;/li&gt; &lt;li&gt; Log periodic fee action for computation {@link Compute} for each of the dates returned above, by calling the service {@link PeriodicFeeLogService.logPeriodicFeeAction}. These actions will be picked up and processed as part of the periodic fee shell. Also in case the periodic fee charged is greater than backdated account opening date then log action.&lt;/li&gt; &lt;li&gt; Call private method {@link this.registerInterestPlanWithBackdate()} to registering interest plan in the interest module with a backdated date(takes value date).&lt;/li&gt; &lt;li&gt; Call private method {@link this.registerForTaxationBackdated()} to register the newly opened account in the Tax module which should now also take account opening date as input.&lt;/li&gt; &lt;li&gt; Call private method {@link this.notifyNPAForBackdatedAccount()} to register the newly opened backdated account in the NPA module which should now also take account opening date as input.&lt;/li&gt;The following are the parameters taken by the method along with their significance&lt;ul&gt; &lt;li&gt; Call private method {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> MinimumAverageBalanceHelper#initializeBackdated AverageBalances(DemandDepositAccount, Date)} to intialize the average balances using the backdated account opening date.&lt;/li&gt; &lt;li&gt; Call {@link EventRateApplicationService#persistRates()} through DDAPricingAdapter to fetch and stamp the net rates for casa interest stages and call {@link WrappedEventRateApplicationService#getLenient RatesByPlanIdWithApplicableUDM()} to derive UDM's and Margin rates and log rate change action.&lt;li&gt;Call updateBundleSubscription of the BundleAccountLinkageService to intimate the bundle subscription unit to update bundle status and benefits, when an account is being opened&lt;/li&gt;&lt;li&gt; call updateCollectionForAccount to intimate collections about the accrual status of the account.&lt;/li&gt;&lt;/ul&gt;&lt;/ol&gt;&lt;/pre&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxationBackdated	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxationBackdated} It accepts following three inputs partyId, accountId, accountBranch,productCode and valueDate which is backdated account opening date. @param partyId @param accountId @param accountBranch @throws FatalException	<ul style="list-style-type: none"> <li>The method is responsible for registering the account with value date which is equivalent to account opening date in the tax module during account opening. It calls {@link TaxManagementService#registerWithTax(depositInformation: DepositDetails)}</li> <li>DepositDetails required in input is populated as follows: <ul style="list-style-type: none"> <li>DepositDetails.partyId = partyId</li> <li>DepositDetails.AccountDepositDetails.accountId = accountId</li> <li>DepositDetails.AccountDepositDetails.ProductClass = DemandDeposit (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.branchCode = accountHomeBranch</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.arrangementNumber = accountId</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.referenceNumber = null</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.ArrangementStatus =Active (This is an enumeration)</li> <li>DepositDetails.AccountDepositDetails.ArrangementDetails.parentArrangementNumber =null</li> </ul> </li> </ul> @param partyId @param accountId @param accountHomeBranch @param productId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxationBackdated	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxationBackdated} It accepts following three inputs partyId, accountid, accountBranch,productCode and valueDate which is backdated account opening date. @param partyId @param accountId @param accountBranch @throws FatalException	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxationBackdated} It accepts following three inputs partyId, accountid, accountBranch,productCode and valueDate which is backdated account opening date. @param partyId @param accountId @param accountBranch @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATaxAdapter	registerForTaxationBackdated	This method uses the DDATaxAdapter to register the newly created account in the tax module. Call {@link IDDATaxAdapter.registerForTaxationBackdated} It accepts following three inputs partyId, accountid, accountBranch,productCode and valueDate which is backdated account opening date. @param partyId @param accountId @param accountBranch @throws FatalException	This method contains the business logic for opening a backdated demand deposit account based on the parameters {@link DemandDepositAccountOpeningDTO} passed by the calling module[eg Origination]. It contains the account opening date which is less than the current process date,it also contains the details of broker/introducer's which was used to open the backdated account. <ol>The following is the algorithm for the method:<li>Fetch all the product level parameters for given offer id {@link OfferInquiryApplicationService#fetchCASAOffer()}<li>Perform all validations using {@link DemandDepositAccountBusinessPolicy#validate()}.<ul>Following validations will be performed in addition to the existing validations of account opening<li>{@link DemandDepositAccountBusinessPolicy#validateBatchRunning()}</li><li>{@link DemandDepositAccountBusinessPolicy#validatePeriodForBackDatedAccount}</li><li>{@link DemandDepositAccountBusinessPolicy#validateOffer



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ForBackDated&lt;/li&gt;&lt;/ul&gt; &lt;li&gt;Generate the account number by using {@link DDAccountOpeningService#generateAccountNumber()}.&lt;/li&gt; &lt;li&gt;Call {@link this.deriveMinorAccountStatus()} to derive the minor status of the primary party and set demandDepositAccount.AccountPreferences.minorAccountStatus = minorAccountStatus&lt;/li&gt; &lt;li&gt;AccountPreferences is a component in the DemandDepositAccount&lt;/li&gt; &lt;li&gt;Call {@link this.setAccountStatus()} to initialize the SystemDefined and UserDefined account statuses&lt;/li&gt; &lt;li&gt;Call private method {@link setAgingStatusDates()} for initializing the aging statuses and dates.&lt;/li&gt; &lt;li&gt; Call {@link setDefaultAccountFacilities()} and {@link setDefaultAccountFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt; Call {@link setDefaultBankingFacilities()} and {@link setDefaultBankingFacilities()} for setting up the default values&lt;/li&gt; &lt;li&gt;Persist the account object to the database by calling {@link DemandDepositAccountRepository#insertDemandDepositAccount()}&lt;/li&gt; &lt;li&gt; Call private method {@link this.createPricingPlanAndEndDialogueForBackdate()} to create pricing plan in the pricing module using the back dated account opening date&lt;/li&gt; &lt;li&gt; Log periodic fee action for computation {@link Compute} for each of the dates returned above, by calling the service {@link PeriodicFeeLogService.logPeriodicFeeAction}. These actions will be picked up and processed as part of the periodic fee shell. Also in case the periodic fee charged is greater than backdated account opening date then log action.&lt;/li&gt; &lt;li&gt; Call private method {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>this.registerInterestPlanWithBackdate()) to registering interest plan in the interest module with a backdated date(takes value date).&lt;/li&gt; &lt;li&gt; Call private method {@link this.registerForTaxationBackdated()} to register the newly opened account in the Tax module which should now also take account opening date as input.&lt;/li&gt; &lt;li&gt; Call private method {@link this.notifyNPABackdatedAccount()} to register the newly opened backdated account in the NPA module which should now also take account opening date as input.&lt;/li&gt;The following are the parameters taken by the method along with their significance&lt;ul&gt; &lt;li&gt; Call private method {@link MinimumAverageBalanceHelper#initializeBackdatedAverageBalances(DemandDepositAccount, Date)} to initialize the average balances using the backdated account opening date.&lt;/li&gt; &lt;li&gt; Call {@link EventRateApplicationService#persistRates()} through DDAPricingAdapter to fetch and stamp the net rates for casa interest stages and call {@link WrappedEventRateApplicationServiceLog#getLenientRatesByPlanIdWithApplicableUDM()} to derive UDM's and Margin rates and log rate change action.&lt;li&gt;Call updateBundleSubscription of the BundleAccountLinkageService to intimate the bundle subscription unit to update bundle status and benefits, when an account is being opened&lt;/li&gt;&lt;li&gt; call updateCollectionForAccount to intimate collections about the accrual status of the account.&lt;/li&gt;&lt;/ul&gt;&lt;/ol&gt;&lt;/pre&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	evaluateAndUpdateTaxWaiver	See {@link DDATaxAdapter#evaluateAndUpdateTaxWaiver} @param accountId @throws FatalException	This method is used update the Account Configurations for a Demand Deposit Account that has already been opened previously. Account level preferences such as statement preferences, minimum balance, debit card preferences, account facilities such as sweep in sweep put instructions, etc. are updated. Additionally, operating instructions both financial and non financial for the account, cheque book requests and default communication address for the account are also updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DDAccountConfigurationDTO,accountConfigurationDTO, , This class represents the DTO class for demand deposit account   configuration.This class contains information about the preferences given for the demand deposit account  as well as preferences captured for each account holder.  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	evaluateAndUpdateTaxWaiver	See {@link DDATaxAdapter#evaluateAndUpdateTaxWaiver} @param accountId @throws FatalException	This method calls Tax module adapter to evaluate and update the tax waiver for the given account id @param sessionContext @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	evaluateAndUpdateTaxWaiver	See {@link DDATaxAdapter#evaluateAndUpdateTaxWaiver} @param accountId @throws FatalException	This method is used to perform different types of registration i.e registration with pricing,interest,NPA and taxation module @param demandDepositClonedAccount @param postingDate @param interestTicketNumber @param parentAccountId @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	evaluateAndUpdateTaxWaiver	See {@link DDATaxAdapter#evaluateAndUpdateTaxWaiver} @param accountId @throws FatalException	In case of regular account opening, we are updating tax waiver as part post account opening activity as it is initiated through origination. See {DDAOriginationAccountApplicationService#updateAccountConfiguration}. In case of cloned account , we are not following origination path it is driven through DDA only. @param sessionContext @param accountId @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	intimateAccountTransferTaxation	@throws FatalException @param partyId @param accountId @param sourceBranch @param productId @throws	Method intimateTaxation will be used to intimate the taxation @param partyId @param accountId @param sourceBranch @param destinationBranch @param productId @throws FatalException
IDDATaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATaxAdapter	unRegisterForTaxation	See {@link DDATaxAdapter#registerForTaxation} @param partyId @param accountId @param accountHomeBranch	This method is common activity which can be called by any service for activities such as 1.validate & delink bundle benefits from the account to be closed by call to bundle 2.close pricing plan by calling pricing 3.close interest plan by calling interest 4.unregister taxation plan by calling taxation 5.remove all the actions on the account 6.Remove all proportioned OD logged from effective date. 7.delink alternate account. @param closureDTO @param account @param agingAccountStatus @param finalBalance @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	initiateCollateralValuation		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId,applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	initiateCollateralValuation		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionDTO} ,collateralDecisionDtos, ,Collateral decision dtos for which the initiation is to be performed @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.InitiateCollateralTasksResponse} This object represents the response that is send with updated collateral decisions after valuation-initiation</p> <p>@exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	initiateTitleSearch		<p>This service initiates title search task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current title search status for the collateral is NOT_INITIATED&lt;br&gt; 2) Title search required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 3) There is no existing unexpired title search report present for the collateral in LCM&lt;br&gt; @param SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	initiateVerificationApplication		This service initiates verification task for application if verification is required by credit decision engine and verification state type is NOT_INITIATED @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @exception  {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013  This exception is thrown if application decision is not found for the given application id.  @throws FatalException
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	expireSubmission		
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processDebitCard		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination ProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processDebitCard		This service process the debit card request , if any, for the given application Id and over drfat product. IfThis service gets called from create account - 1 pg serial number will be passed in the array input if called form settlement all the pg's in the application will be passed in the array input @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String applicationId @fcb.param MI,Integer[] product group serial numbers @return DebitCardResponse, , The response contains the debit card details @throws FatalException
IOrigination ProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	terminateSubmission		
IOrigination ProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processSubmission		
IOrigination ProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processSubmission		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processSubmission		
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	markCreditAssessmentProcessCompletion		<p>This method checks if post credit assessment activities have been completed for the given submission id &lt;br&gt; Method checks for application verification for all lending applications along with collateral valuation &lt;br&gt; and title search completion status for all the collaterals in the submission &lt;br&gt; If credit decision has been done then ConductCreditAssessment BPEL process is called to complete &lt;br&gt; the credit assessment process&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CreditDecisionCompletionResponse} &lt;br&gt; A response to indicate if the credit decision has been completed @exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if credit decision does not exist for the input submission. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processOpenCreditCardAccount		This service is used to do post verification operation.  For credit card application, this checks for the structure solution confirmation status. If all product groups are confirmed, then invokes the process to process Open Credit Card Account @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param String,applicationId, ,applicationId Unique Identifier for the application @return TransactionStatus, transactionStatus, , Transaction status of the operation   @throws FatalException
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processOpenCreditCardAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	initiateCollateralValuations		<p>This service initiates valuation task for all collaterals linked to given application. The following conditions are checked before initiating the task-&lt;br&gt; 1) Current valuation status for the collateral is NOT_INITIATED&lt;br&gt; 2) Valuation is required is returned by the decision engine for the collateral decision 3) Valuation required is maintained for the collateral category in LCM Collateral Preferences&lt;br&gt; 4) There is no existing unexpired valuation report present for the collateral and valuation state type in LCM&lt;br&gt; @param SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.CollateralDecisionDTO} ,collateralDecisionDtos, ,Collateral decision dtos for which the initiation is to be performed @return {@link com.ofss.fc.app.origination.dto.lending.core.credit.decision.InitiateCollateralTasksResponse} This object represents the response that is send with updated collateral decisions after valuation-initiation</p> <p>@exception&lt;br&gt; {@link OriginationErrorConstants#OR_SUBMISSION_DECISION_NOT_FOUND} : FC_OR_CD_0012&lt;br&gt; This exception is thrown if submission decision is not found for the given submission id.&lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_DECISION_NOT_FOUND} : FC_OR_CD_0013&lt;br&gt; This exception is thrown if application decision is not found for the given application id.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processInitializeMCD		<p>This service is called when the "Invoke MCD" button in the app tracker or the Decline Offer Task is clicked for AUTO declined submission. &lt;br&gt; (1) It aborts the fault task flow instance&lt;br&gt; (2) It will initialize the MCD task for Submissions, if the submission decision is at DecisionStageType.AUTO and decision status type is DecisionStatusType.DECLINE. It will throw the exception {@link OriginationErrorConstants#OR_NOT_ELIGIBLE_FOR_MCD} if this condition is not satisfied. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc @fcb.param MI,String,submissionId, ,submissionId Unique Identifier for the submission @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. &lt;br&gt; @exception&lt;br&gt; {@link OriginationErrorConstants#OR_NOT_ELIGIBLE_FOR_MCD} : FC_OR_CD_0034&lt;br&gt; This exception is thrown this submission is not eligible for MCD.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationProcessAdapter	processInitializeMCDForVariationRequest		This service is called when the "Invoke MCD" button in the app tracker or the Decline Offer Task is clicked for AUTO declined submission.   (1) It aborts the fault task flow instance  (2) It will initialize the MCD task for Submissions, if the submission decision is at DecisionStageType.AUTO and decision status type is DecisionStatusType.DECLINE. It will throw the exception {@link OriginationErrorConstants#OR_NOT_ELIGIBLE_FOR_MCD} if this condition is not satisfied. @param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc @param MI,String,submissionId, submissionId Unique Identifier for the submission @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.   @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	This method publishes warning indicator changes to collection. @param sessionContext @param partyWarningIndicatorDTO @param memoList @param modifiedPartyWarningIndicatorDTO @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	This method publishes warning indicator changes to collection. @param sessionContext @param partyWarningIndicatorDTO @param memoList @param modifiedPartyWarningIndicatorDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method is used to execute the records picked up on the date of the batch processing during the statement category processing. The partyId comes in the {@link PartyBasedActionKey} and the determinant value and the status are deserialized from the extra attributes. This method updates the party status to 'Operational Major' or 'Legal Major' whichever is applicable. It also updates the statuses of the accounts under the given partyId.
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in DDA Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in TD Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status. If the status is updated to 'Operational Major' then another action is logged for updating the status to 'Legal Major' when the party reaches the legal major age. This action is picked up and processed during the batch run on the pick up date logged. @param sessionContext the SessionContext @param partyId the party Id @param determinantValue the determinantValue for the party @param newStatus the new Status of the party @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method is used to modify individual party demographics details. @fcb.param MI,SessionContext,sessionContext, , This parameter captures   the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,CorporateDemographicsDTO partyDemographicsDTO  CorporateDemographicsDTO contains following inputs  @fcb.param MI, partyType, Indicate the legal entity or the banking entity. @fcb.param MI, partyId, Indicate the party id whose demographics will be updated. @fcb.param MI, maidenNameOfMother @fcb.param MI, branch, Indicates the home branch of the party. @fcb.param MI, privacyOptout, @fcb.param MI, shareInformationIndicator, @fcb.param MI, gender=MAL @fcb.param MI, birthDate, @fcb.param MI, birthPlace, @fcb.param MI, isNonResident=false @fcb.param MI, isPrivacyOptout=false @fcb.param MI, maritalStatus=MAR @fcb.param MI, maritalStatusAsOfDate=20070213 @fcb.param MI, incomeCategoryId=1 @fcb.param MI, languageCode=ENG @fcb.param MI, ethnicity=Indian @fcb.param MI, race=Indian @fcb.param MI, religion=Indian @fcb.param MI, nationality=INDIAN @fcb.param MI, domicile=ORISSA @fcb.param MI, countryOfOriginCode=IND @fcb.param MI, PhysicalStatusDTO.isDisabled=false @fcb.param MI, PhysicalStatusDTO.dateOfDemise= @fcb.param MI, PhysicalStatusDTO.heightInCentimetres=168 @fcb.param MI, PhysicalStatusDTO.diseases=leg @fcb.param MI, PhysicalStatusDTO.description=leg @fcb.param MI, PhysicalStatusDTO.bloodType=O+

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@fcb.param MI,PhysicalStatusDTO.smokingIndicator=false <ul> Personal Interests <li>@fcb.param MI,PersonalInterest</li> </ul> <ul> Dependents <li>@fcb.param MI,serialNumber, <li>@fcb.param MI,name, <li>@fcb.param MI,age, <li>@fcb.param MI,relationCode, </ul> <ul> educationProfileDTO <li>@fcb.param MI,educationType, <li>@fcb.param MI,yearOfDegree, <li>@fcb.param MI,universityName, <li>@fcb.param MI,educationMode,
IPartyStatus ChangeAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyStatusCha ngeAdapter	updatePart yStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	
IPartyStatus ChangeAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyStatusCha ngeAdapter	updatePart yStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status @param sessionContext the SessionContext @param partyId the party Id @param determinantValue the determinantValue for the party @param newStatus the new Status of the party @throws FatalException
IPartyStatus ChangeAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyStatusCha ngeAdapter	updatePart yStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in DDA Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatus ChangeAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyStatusCha ngeAdapter	updatePart yStatus	This method updates party status in different modules. Basically,this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in TD Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method is the service method for the 'pi_status_chg_shell' shell. This method will be called by the Batch Framework to notify other modules about the Party's status change. The modules that are currently notified by this method are DDA & TD. @param context the ApplicationContext @param partyStatusChangeBatchData the BatchData, corresponding to a single record in the 'pi_status_chg_shell' shell's driver table
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in DDA Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status in TD Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updatePartyStatus	This method updates party status in different modules. Basically, this requirement arises due to party status change. @param accountId @param newStatus	This method will update the party Status @param sessionContext the SessionContext @param partyId the party Id @param determinantValue the determinantValue for the party @param newStatus the new Status of the party @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	fetchAccountStatus	This method fetches Account minority status from different modules. @param accountId @return String	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	fetchAccountStatus	This method fetches Account minority status from different modules. @param accountId @return String	
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	fetchAccountStatus	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the party Status in TD Module @param sessionContext the SessionContext @param partyId the party Id @param newStatus the new Status of the party
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This represents the updatePartyWarningIndicator method of the entity. This method gets called whenever warning indicator is linked or delinked from party. It also updates the warning indicator changes on party to other modules where it has an impact. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,{@link PartyWarningIndicatorDTO},partyWarningIndicatorDTO PartyWarningIndicatorDTO contains the following: partyId: The warning indicators will be set for this party id. partyType: Mandatory Input This will indicate if the party is individual, organization or banking entity. warningIndicatorId: The type of warning Indicator. isApplicable: It indicates whether the warning indicator is set or not. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This represents the updatePartyWarningIndicator method of the entity. This method gets called whenever warning indicator is linked or delinked from party. It also updates the warning indicator changes on party to other modules where it has an impact. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,{@link PartyWarningIndicatorDTO},partyWarningIndicatorDTO PartyWarningIndicatorDTO contains the following: partyId: The warning indicators will be set for this party id. partyType: Mandatory Input This will indicate if the party is individual, organization or banking entity. warningIndicatorId: The type of warning Indicator. isApplicable: It indicates whether the warning indicator is set or not. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator changes in DDA Module. If the warning indicator added to party has impact on DDA accounts and party has DDA accounts then it will update the warning indicator impact on DDA accounts of party @param existingPartyWarningIndicator @param sessionContext the SessionContext @param partyId the partyWarningIndicator

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator changes in DDA Module. If the warning indicator added to party has impact on DDA accounts and party has DDA accounts then it will update the warning indicator impact on DDA accounts of party @param existingPartyWarningIndicator @param sessionContext the SessionContext @param partyId the partyWarningIndicator
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	This method will update the Warning Indicator in Loans Module @param sessionContext the SessionContext @param partyId the partyWarningIndicator @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	publishWarningIndicator	This method fetches Account minority status from different modules. @param accountId @return String	
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updateCollectionFlagForPartyAccounts	This method is added to modify the isCollection flag in accounts. @param sessionContext @param partyId @throws FatalException	This method updates Collection Flag For Party Accounts. @param sessionContext @param partyId @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updateCollectionFlagForPartyAccounts	This method is added to modify the isCollection flag in accounts. @param sessionContext @param partyId @throws FatalException	This method updates Collection Flag For Party Accounts. @param sessionContext @param partyId @throws FatalException
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updateCollectionFlagForPartyAccounts	This method is added to modify the isCollection flag in accounts. @param sessionContext @param partyId @throws FatalException	This method is added to take action against any specific warning indicators.   <b>Action 1.</b> When the Warning Indicator defined as 'Deceased' then service provider status is also   marked accordingly   <b>Action 2.</b> When the Warning Indicator defined as 'Recovery', the IS_ACCOUNT_IN_COLLECTION flag in the table FLX_LN_ACCT_ADDN_DETLS_B has to be modified to Y. Through the PartyStatusChangeAdapter, the updateInCollections() method is called in LoanAccountCollectionInformationApplicationService to update the same. @param warningIndicatorId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyStatusChangeAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyStatusChangeAdapter	updateCollectionFlagForPartyAccounts	This method is added to modify the isCollection flag in accounts. @param sessionContext @param partyId @throws FatalException	This method is added to take action against any specific warning indicators.   <b>Action 1.</b> When the Warning Indicator defined as 'Deceased' then service provider status is also   marked accordingly   <b>Action 2.</b> When the Warning Indicator defined as 'Recovery', the IS_ACCOUNT_IN_COLLECTION flag in the table FLX_LN_ACCT_ADDN_DETLS_B has to be modified to Y. Through the PartyStatusChangeAdapter, the updateInCollections() method is called in LoanAccountCollectionInformationApplicationService to update the same. @param warningIndicatorId @throws FatalException
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchSimulatedClosureLogsForTheAccountId		
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchSimulatedClosureLogsForTheAccountId		This class is used for fetching the simulated logs which were persisted during the account closure simulation @param sessionContext @param accountId @throws FatalException
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchFutureDatedClosureDetails		This method is used for fetching the account details in case of future dated account closure simulation  This will take the account id and the expected closure date in the input  It will call the simulation service for generating the simulated response @throws FatalException
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchFutureDatedClosureDetails		This method is used for fetching the account details in case of future dated account closure simulation  This will take the account id and the expected closure date in the input  It will call the simulation service for generating the simulated response



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchFutureDatedClosureDetails		This method is used for fetching the account details in case of future dated account closure simulation  This will take the account id and the expected closure date in the input  It will call the simulation service for generating the simulated response
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	fetchFutureDatedClosureDetails		This will return the simulated state for the given simulation id.  The state of the account has been persisted in the database during the simulation  The exact state at that time would be returned to the caller
IDDASimulationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASimulationAdapter	updateRecordToBeDisplayedFlag		This method is exposed to updated the waived amount during the account closure  @param sessionContext @param simulationId @param ddaClosureRequestDTO @throws FatalException
IInterestTDApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTDApplicationAdapter	fetchLienAmountOnInterest	Responsible for fetching the lien amount on a TD account	This method will fetch the compounded interest amount available for adjustment. @param interestPlanItem @return
IInterestTDApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestTDApplicationAdapter	fetchLienAmountOnInterest	Responsible for fetching the lien amount on a TD account	This method will fetch the compounded interest amount available for adjustment. @param interestPlanItem @return
IOriginationEnumerationManagerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationEnumerationManagerAdapter	fetchOpenEnumerationDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination Enumeration ManagerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationEnumerationManagerAdapter	fetchOpenEnumerationDetails		This method provides the implementation for validating the annotation {@link Enum}. This class invokes <i>Enumeration Manager</i> via <i>adapter</i>. It then validates the response returned by the Enumeration Manager against the field to which the annotation is attached, if there is a miss-match then respective error code (as configured in {@link Enum} annotation) is thrown.
IOrigination Enumeration ManagerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationEnumerationManagerAdapter	fetchOpenEnumerationDetails		
IBundleParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundlePartyAdapter	isPartyInHardship	<p>This method will determine if the input party is in hardship. </p> @fcb.param MI, String, partyId, Indicates the party which is to be verified for hardship. @return @throws FatalException	
IBundleParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundlePartyAdapter	isPartyInHardship	<p>This method will determine if the input party is in hardship. </p> @fcb.param MI, String, partyId, Indicates the party which is to be verified for hardship. @return @throws FatalException	
IBundleParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundlePartyAdapter	isPartyInHardship	<p>This method will determine if the input party is in hardship. </p> @fcb.param MI, String, partyId, Indicates the party which is to be verified for hardship. @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundlePartyAdapter	isPartyInHardship	<p>This method will determine if the input party is in hardship. </p> @fcb.param MI, String, partyId, Indicates the party which is to be verified for hardship. @return @throws FatalException	<p>This method will validate of the input list of bundle owners are in hardship. </p> @param bundleOwner @return
IBundleParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundlePartyAdapter	isPartyInHardship	<p>This method will determine if the input party is in hardship. </p> @fcb.param MI, String, partyId, Indicates the party which is to be verified for hardship. @return @throws FatalException	<p>This method will validate of the input list of bundle owners are in hardship. </p> @param bundleOwner @return
IBranchProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchProductAdapter	fetchLedgerCode		
IBranchProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchProductAdapter	fetchLedgerCode		This method will fetch ledger code
IBranchProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchProductAdapter	fetchLedgerCode		This method will transfer money to GL from CASA account
IBranchProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchProductAdapter	fetchLedgerCode		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IUdfServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\udf\IUdfServiceAdapter	addLinkedUDF	@param sessionContext @param linkedUDFDTO @param dtoValueMap @return @throws FatalException	
IUdfServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\udf\IUdfServiceAdapter	addLinkedUDF	@param sessionContext @param linkedUDFDTO @param dtoValueMap @return @throws FatalException	This method is used to add UDF log details for a task code. The task code is linked to one of the three categories, on the basis of which the UDFs are logged. These three categories are, 1) Customer (for which Customer Id must be given) 2) Account (for which Account Id must be given) 3) Task (for which External Reference Number must be given in the SessionContext) If the task or transaction is NOT of OLTP type, then the Field Reference Number must be given. @fcb.param MI, SessionContext,sessionContext, which must contain the session ticket and user id.  @fcb.param MI, FieldDetail[], fieldDetail, array containing the names and values of the UDFs.  @fcb.param MI, String,taskCode, the transaction mnemonic for which the UDFs are to be added.  @return TransactionStatus, status, The status of the transaction which indicates success or failure. <b> Transaction failure</b> The error codes will indicate the failure of the transactions   <ul> <li> {@link com.ofss.fc.common.FAPIErrorConstants#MID_ESTMT_INQ_UDF_DET_NOT_FOUND}</li> <li> {@link com.ofss.fc.common.FAPIErrorConstants#MID_MANDATORY_FIELD_NULL}</li> <li> {@link com.ofss.fc.common.FAPIErrorConstants#MID_BLANK_FIELD}</li> <li> {@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_NO_ROWS}</li> <li> {@link com.ofss.fc.common.FAPIErrorConstants#MID_INV_I

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> <a href="#">NPUT}</a></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}</a></li> <li><a href="#">{@link com.ofss.fc.common.FAPIErrorConstants#MID_ECI_C UST_INVALID}</a></li> <li><a href="#">{@link com.ofss.fc.common.FAPIErrorConstants#MID_ECI_I NV_CUST_ID}</a></li> </ul> <p> @throws FatalException  Please refer to the documentation above for  <i>Transaction failure</i> section @see also  com.ofss.fc.entity.udf.log.FieldDetail</p>
IUdfServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\udf\IUdfServiceAdapter	updateLinkedUDF	@param sessionContext @param linkedUDFDTO @param dtoValueMap @return @throws FatalException	
IUdfServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\udf\IUdfServiceAdapter	updateLinkedUDF	@param sessionContext @param linkedUDFDTO @param dtoValueMap @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyDetails	@param partyID @return	This method used to fetch the details of that party. This will call {@link ITaxationPartyApplicationAdapter} adapter interface to fetch the details by passing party id as input. @param partyID @return PartyInformation
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyDetails	@param partyID @return	This method will fetch the {@link PartyInformation} based on the party id.  This will make a call to {@link PartyApplicationService#fetchPartyBasicDetails(SessionContext, String)} to fetch the party information.  The above service will return {@link PartyInquiryResponse}.  With the response object it will make {@link PartyInformation} response object. @param partyID @return partyInformation
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyDetails	@param partyID @return	@param accountID @param depositInformationDTO @param taxParameterDTO @return @throws FatalException
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyDetails	@param partyID @return	@param accountID @param processDate @param daysForTaxCalculation @param dayCountConventionType @param taxParameterDTO @return @throws FatalException
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	getPartyRelationDetailsForAccount	@param accountId @return	@param accountID @param depositInformationDTO @param taxParameterDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																								
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	getPartyRelationDetailsForAccount	@param accountId @return	@param accountID @param processDate @param daysForTaxCalculation @param dayCountConventionType @param taxParameterDTO @return @throws FatalException																								
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	getAccountRelationDetailsForparty																										
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	getAccountRelationDetailsForparty		<p>This service is used to fetch the tax waiver maintained for an arrangement . This method takes the {@link TaxWaiver} array as input. &lt;br&gt;This method will load the {@link DepositInformation} by calling {@link TaxManagementService#loadDepositDetails}.&lt;br&gt;This will validate both {@link TaxWaiver} and {@link DepositInformation} domain objects. &lt;br&gt; If any new deposit found in the {@link DepositInformation} then it will concatenate with {@link TaxWaiver} and return it .&lt;br&gt; For example If a party has three accounts and in that two are CASA accounts and one is TD account.Then the following information will be stored in {@link DepositInformation}.&lt;br&gt;</p> <table border="1"> <thead> <tr> <th>Party ID</th> <th>Account ID</th> <th>ProductClass</th> <th>Arrangement Number</th> <th>Reference Number</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>A1</td> <td>CASA</td> <td>A1</td> <td>1234561</td> <td>Active</td> </tr> <tr> <td>P2</td> <td>A2</td> <td>CASA</td> <td>A2</td> <td>1234562</td> <td>Active</td> </tr> <tr> <td>P3</td> <td>A3</td> <td>TD</td> <td>A3#D1</td> <td>1234563</td> <td></td> </tr> </tbody> </table>	Party ID	Account ID	ProductClass	Arrangement Number	Reference Number	Status	P1	A1	CASA	A1	1234561	Active	P2	A2	CASA	A2	1234562	Active	P3	A3	TD	A3#D1	1234563	
Party ID	Account ID	ProductClass	Arrangement Number	Reference Number	Status																							
P1	A1	CASA	A1	1234561	Active																							
P2	A2	CASA	A2	1234562	Active																							
P3	A3	TD	A3#D1	1234563																								

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Active&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; If waiver is maintaining for the first time then it will show all three accounts on the screen from which teller is maintaining the waiver.Teller has maintained waiver for all then the following information will be stored. &lt;table class="tableizer- table"&gt; &lt;tr class="tableizer-firststrow"&gt; &lt;th&gt;Party ID&lt;/th&gt; &lt;th&gt;Account ID&lt;/th&gt; &lt;th&gt;Deposit Number&lt;/th&gt; &lt;th&gt;Waiver Criteria&lt;/th&gt; &lt;th&gt;Waiver Reason&lt;/th&gt; &lt;th&gt;Waiver Level&lt;/th&gt; &lt;th&gt;Financial Year&lt;/th&gt; &lt;th&gt;Waiver For all&lt;/th&gt; &lt;th&gt;Product Class&lt;/th&gt; &lt;th&gt;Waiver Start Date&lt;/th&gt; &lt;th&gt;Waiver End Date&lt;/th&gt; &lt;th&gt;isTaxWaived&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A1&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A2&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A3&lt;/td&gt; &lt;td&gt;D1&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;TD&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; Later customer opened one more term deposit under same old TD account.&lt;br&gt; Then the following information will be stored. &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer-firststrow"&gt; &lt;th&gt;Party ID&lt;/th&gt; &lt;th&gt;Account ID&lt;/th&gt; &lt;th&gt;ProductClass&lt;/th&gt; &lt;th&gt;Arrangement Number&lt;/th&gt; &lt;th&gt;Reference </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> Number&lt;/th&gt; &lt;th&gt;Status&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A1&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;A1&lt;/td&gt; &lt;td&gt;1234561&lt;/td&gt; &lt;td&gt;Active&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A2&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;A2&lt;/td&gt; &lt;td&gt;1234562&lt;/td&gt; &lt;td&gt;Active&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A3&lt;/td&gt; &lt;td&gt;TD&lt;/td&gt; &lt;td&gt;A3#D1&lt;/td&gt; &lt;td&gt;1234563&lt;/td&gt; &lt;td&gt;Active&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A3&lt;/td&gt; &lt;td&gt;TD&lt;/td&gt; &lt;td&gt;A3#D2&lt;/td&gt; &lt;td&gt;1234564&lt;/td&gt; &lt;td&gt;Active&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; If teller inquire the waiver details then it will give following information. &lt;style type="text/css"&gt; table.tableizer-table {border: 1px solid #CCC; font- family: Arial, Helvetica, sans-serif; font-size: 12px;} .tableizer-table td {padding: 4px; margin: 3px; border: 1px solid #ccc;} .tableizer-table th {background-color: #104E8B; color: #FFF; font-weight: bold;} &lt;/style&gt; &lt;table class="tableizer-table"&gt; &lt;tr class="tableizer- firstrow"&gt; &lt;th&gt;Party ID&lt;/th&gt; &lt;th&gt;Account ID&lt;/th&gt; &lt;th&gt;Deposit Number&lt;/th&gt; &lt;th&gt;Waiver Criteria&lt;/th&gt; &lt;th&gt;Waiver Reason&lt;/th&gt; &lt;th&gt;Waiver Level&lt;/th&gt; &lt;th&gt;Financial Year&lt;/th&gt; &lt;th&gt;Waiver For all&lt;/th&gt; &lt;th&gt;Product Class&lt;/th&gt; &lt;th&gt;Waiver Start Date&lt;/th&gt; &lt;th&gt;Waiver End Date&lt;/th&gt; &lt;th&gt;isTaxWaived&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A1&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A2&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;CASA&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A3&lt;/td&gt; &lt;td&gt;D1&lt;/td&gt; &lt;td&gt;TFN&lt;/td&gt; &lt;td&gt;TFN submitted&lt;/td&gt; &lt;td&gt;Account&lt;/td&gt; &lt;td&gt;2010&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;TD&lt;/td&gt; &lt;td&gt;19/10/2010&lt;/td&gt; &lt;td&gt;31/03/2011&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;P1&lt;/td&gt; &lt;td&gt;A3&lt;/td&gt; &lt;td&gt;D2&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&amp;nbsp;&lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; @param taxWaiver </pre>
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	getAccountRelationDetailsForparty		This method is used to fetch the list of active account numbers i.e either CASA or TD which belongs to a particular party The user can fetch the arrangement numbers which belongs to Single branch or Multiple branches @param partyID @param productClass @param branchName @param taxYear @return List of TaxWaiver
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyExemptionDetails		This method is used to retrieve the exemption limit. The following algorithm used to fetch the exemption limit. <li>Call party module to fetch the list of {@link PartyAccountRelationshipDetails} of that account by passing account number through adapter. <li>Call {@link this#fetchPartyDetails(String)} to fetch the {@link PartyInformation} by passing party id. <li>Call {@link this#evaluateExemptionLimit(String, PartyInformation, Date)} by passing rule id, {@link PartyInformation} and process date. <li>Read total exemption limit and prorate frequency. <li>Based on the prorate frequency compute exemption limit. <li>Repeat the steps 2 to 4 and fetch @param partyID @param taxParameter @param processDate @param depositInformation @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyExemptionDetails		This method is used to retrieve the exemption limit. The following algorithm used to fetch the exemption limit. <li>Call party module to fetch the list of {@link PartyAccountRelationshipDetails } of that account by passing account number through adapter. <li>Call {@link this#fetchPartyDetails(String)} to fetch the {@link PartyInformation} by passing party id. <li>Call {@link this#evaluateExemptionLimit(String, PartyInformation, Date)} by passing rule id, {@link PartyInformation} and process date. <li>Read total exemption limit and prorate frequency. <li>Based on the prorate frequency compute exemption limit. <li>Repeat the steps 2 to 4 and fetch @param partyID @param taxParameter @param processDate @param depositInformation @return
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyExemptionDetails		This method is used to retrieve the exemption limit. The following algorithm used to fetch the exemption limit. <li>Call party module to fetch the list of {@link PartyAccountRelationshipDetails } of that account by passing account number through adapter. <li>Call {@link this#fetchPartyDetails(String)} to fetch the {@link PartyInformation} by passing party id. <li>Call {@link this#evaluateExemptionLimit(String, PartyInformation, Date)} by passing rule id, {@link PartyInformation} and process date. <li>Read total exemption limit and prorate frequency. <li>Based on the prorate frequency compute exemption limit. <li>Repeat the steps 2 to 4 and fetch @param partyID @param taxParameter @param processDate @param depositInformation @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationPartyApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationPartyApplicationAdapter	fetchPartyExemptionDetails		This method is used to retrieve the exemption limit. The following algorithm used to fetch the exemption limit. <li>Call party module to fetch the list of {@link PartyAccountRelationshipDetails } of that account by passing account number through adapter. <li>Call {@link this#fetchPartyDetails(String)} to fetch the {@link PartyInformation} by passing party id. <li>Call {@link this#evaluateExemptionLimit(String, PartyInformation, Date)} by passing rule id, {@link PartyInformation} and process date. <li>Read total exemption limit and prorate frequency. <li>Based on the prorate frequency compute exemption limit. <li>Repeat the steps 2 to 4 and fetch @param partyID @param taxParameter @param processDate @param depositInformation @return
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate SubLine for Main Line of Product Groups. @param context @param facilityId @param loanAccountSublineDTO @param outboundDocumentRequestDTO @return SubLine @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate SubLine for Main Line of Product Groups. @param context @param facilityId @param loanAccountSublineDTO @param outboundDocumentRequestDTO @return SubLine @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate MainLine for product Group of Offer Letter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate SubLine for Main Line of Product Groups. @param context @param facilityId @param loanAccountSublineDTO @param outboundDocumentRequestDTO @return SubLine @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method is used to populate LMIDetails for MainLine of Product Groups. @param context @param facilityId @return LMIDetails @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchFacilityHierarchy	This method is used to get all the child or sub facilities under a given main facility the selection will be on a view that queries both proposed and actual tables the records are filtered based on the rank @param sessionContext @param facilityId @param considerProposed @return FacilityInquiryResponse @throws FatalException	This method populates the Collateral attributes for the offerLetter extract. It performs following steps: <p> 1.Calls ProposedFacilityInquiryApplicationService.fetchCollateralsForProposedFacility on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityResponse @param applicationId @param facilityId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchAccountsForFacility	This method is used to obtain the accounts that are linked to the credit Facility. @param sessionContext @param creditFacilityId @throws FatalException	
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchAccountsForFacility	This method is used to obtain the accounts that are linked to the credit Facility. @param sessionContext @param creditFacilityId @throws FatalException	This method is used to obtain the accounts that are linked to the credit Facility. @param sessionContext @param creditFacilityId @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchInsuranceLinkagesforFacility	This method used to get the insurance linkages for the given facility ID and policy number @param sessionContext @param facilityID @param policyNO @return @throws FatalException	This method used to get the insurance linkages for the given facility ID and policy number @param sessionContext @param facilityID @param policyNO @return @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchInsuranceLinkagesforFacility	This method used to get the insurance linkages for the given facility ID and policy number @param sessionContext @param facilityID @param policyNO @return @throws FatalException	This method is used to populate LMIDetails for MainLine of Product Groups. @param context @param facilityId @return LMIDetails @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityInfo	This method is used to obtain the credit facility details for the given Credit facility. This method will update CreditFacility in FacilityInquiryResponse and return the response @param sessionContext @param creditFacilityId @throws FatalException	This method is used to obtain the credit facility details for the given Credit facility. This method will update CreditFacility in FacilityInquiryResponse and return the response @param sessionContext @param creditFacilityId @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityCovenantLinkages	@param session @param borrowingEntity @param facilityId @return FacilityCovenantLinkageInquiryResponse @throws FatalException	@param session @param borrowingEntity @param facilityId @return FacilityCovenantLinkageInquiryResponse @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchFacilityConditionLinkages	@param session @param borrowingEntity @param facilityId @return FacilityConditionLinkageInquiryResponse @throws FatalException	@param session @param borrowingEntity @param facilityId @return FacilityConditionLinkageInquiryResponse @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchCollateralByBorrowingEntity	@param sessionContext @param borrowingEntityId @throws FatalException	@param sessionContext @param borrowingEntityId @throws FatalException
IntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationLCMAAdapter	fetchCollateralDetails	@param sessionContext @param collateralId @throws FatalException	@param sessionContext @param collateralId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IIIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IIIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	
IIIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the MortgageLetter extract. It performs following steps: <p> 1.Calls FacilityInquiryApplicationService.fetchFacilityInfo on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Calls CollateralApplicationService.fetchCollateralByBorrowingEntity <p> 4.Populates the Collateral related attributes @param SessionContext @param facilityId @return Collateral @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the offerLetter extract. It performs following steps: <p> 1.Calls ProposedFacilityInquiryApplicationService.fetchCollateralsForProposedFacility on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityResponse @param applicationId @param facilityId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the MortgageLetter extract. It performs following steps: <p> 1.Calls FacilityInquiryApplicationService.fetchFacilityInfo on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Calls CollateralApplicationService.fetchCollateralByBorrowingEntity <p> 4.Populates the Collateral related attributes @param SessionContext @param facilityId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchCollateralsForFacilityIdList	@param sessionContext @param facilityId @return CollateralDTO @throws FatalException	This method populates the Collateral attributes for the offerLetter extract. It performs following steps: <p> 1.Calls ProposedFacilityInquiryApplicationService.fetchCollateralsForProposedFacility on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityResponse @param applicationId @param facilityId @return Collateral @throws FatalException
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchProposedConditionLinkagesForBorrowingEntity	@param sessionContext @param borrowingEntity @return @throws FatalException	
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchProposedCovenantLinkagesForBorrowingEntity	@param sessionContext @param borrowingEntity @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchLRMPolicyDetailsForApplicationId	The method is called when the user wants to view the basic details of the LMI policy given the ApplicationId.  @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, Indicates the application Id @return LRMPolicyDetailsResponse @exception OriginationErrorConstants.OR_INSURANCE_NOT_FOUND : FC_OR_MI_0002   This exception is thrown when we try to fetch the policy details   for an application for which policy has not been initiated.	The method is called when the user wants to view the basic details of the LMI policy given the ApplicationId.  @return LRMPolicyDetailsResponse
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchBasicLRMPolicyDetails	The method is called when the user wants to view the basic details of the LMI policies given the submissionId and FacilityId. This service is provided for the LMI interface for their inquiry. @return LRMPolicyDetailsResponse	The method is called when the user wants to view the basic details of the LMI policy given the FacilityId.  @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,String facilityId, Indicates the facility Id @return LRMPolicyDetailsResponse
IIntegrationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationLCMAAdapter	fetchBasicLRMPolicyDetails	The method is called when the user wants to view the basic details of the LMI policies given the submissionId and FacilityId. This service is provided for the LMI interface for their inquiry. @return LRMPolicyDetailsResponse	This method is used to populate LMIDetails for MainLine of Product Groups. @param context @param facilityId @return LMIDetails @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	assertFraudPolicies		
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	assertFraudPolicies		This method loads resource for service and returns TransactionCheckpoint. @return String @throws FatalException
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	assertFraudPolicies		
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	assertFraudPolicies		
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	updateTransactionStatus		
IFraudAssertionWebServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionWebServiceAdapter	updateTransactionStatus		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationRuleAdapter	executeRule	Method will be used to execute rule with given rule id and fact map for the effective date @param ruleId @param factsMap @return	
IAssetClassificationRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationRuleAdapter	executeRule	Method will be used to execute rule with given rule id and fact map for the effective date @param ruleId @param factsMap @return	This method calls the Rule module via RuleAdapter to execute the asset classification rule to derive the expected classification code @param ruleId @param factsInRule @param facilityFacts @param arrearsDetails @return @throws FatalException
IAssetClassificationRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationRuleAdapter	executeRule	Method will be used to execute rule with given rule id and fact map for the effective date @param ruleId @param factsMap @return	This method is called by the facility batch service to derive the expected code for the given main line1 @param ruleId @param factsInRule @param facilityBatchFacts @param forNextClassificationDate @param arrearsDetails @return @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	undoImmediatecredit	@param accountTransactionDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	undoImmediatecredit	@param accountTransactionDTO @return @throws FatalException	This method is called on reversal of immediate credit transaction. @param accountTransactionDTO @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	undoImmediatecredit	@param accountTransactionDTO @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchChequeStatus	@param accNumber @param chequeNumber @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchChequeStatus	@param accNumber @param chequeNumber @return @throws FatalException	This method will check if the instrument.getInstrumentNumber() belongs to instrument.getCounterPartyAccount() by delegating the call to CASA Cheque Service. If not, it will raise validation exception @param instrument @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchChequeStatus	@param accNumber @param chequeNumber @return @throws FatalException	@param chequeStatusInquiryDTO @param chequeStatus @param IPCCASAAadapter @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchChequeStatus	@param accNumber @param chequeNumber @return @throws FatalException	@param chequeSummarySet @param statusResponse @param flag @param IPCCASAAadapter @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	verifyAccount	@param accountId @param amount @param serviceId @param eventId TODO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	verifyAccount	@param accountId @param amount @param serviceId @param eventId TODO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	verifyAccount	@param accountId @param amount @param serviceId @param eventId TODO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	verifyAccount	@param accountId @param amount @param serviceId @param eventId TODO @return @throws FatalException	@param logItem @param amount @param txnlog
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo	This method will perform credit to the CASA account for which business payment was initiated, in case an error is encountered during procurement of funds or perform debit from the CASA account for which business collection was initiated, in case an error is encountered while debiting an account which belongs to the same bank.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.buinesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, boolean, isFirstCall <li>Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.</li> @return AccountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	creditFunds	This method will credit the savings account. Although cheque no is provided, it does not put the amount in uncleared balance. For uncleared balance, a separate operation has to be invoked. It should also generate a entry in account audit trail @param context @param accountID @param amount @param chequeNo	@param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateBalance	@param pcAccountTransactionDTO @param creditRequestDTO @return @throws FatalException	This method will perform credit to the CASA account for which business payment was initiated, in case an error is encountered during procurement of funds or perform debit from the CASA account for which business collection was initiated, in case an error is encountered while debiting an account which belongs to the same bank.   @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, IBulkTransferInstruction, bulkTransferInstruction <li> {@link com.ofss.fc.domain.pc.businesstransfer.entity.IBulkTransferInstruction}</li> @fcb.param MI, FundTransferType, fundTransferType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @fcb.param MI, boolean, isFirstCall <li>Indicates if it is first call, then attempt to reverse to trace account, else attempt to reverse to provider account.</li> @return AccountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateBalance	@param pcAccountTransactionDTO @param creditRequestDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateBalance	@param pcAccountTransactionDTO @param creditRequestDTO @return @throws FatalException	This service validate the policy of immediate credit as well as perform the Immediate credit on provided data. It will raise an accounting event for the credit and also audit the transaction @throws FatalException @see com.ofss.fc.app.pc.core.IOutgoingCollectionService#doImmediateCredit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.dto.log.InstrumentTransferTransactionLogDTO, com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO, com.ofss.fc.datatype.Money)
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateBalance	@param pcAccountTransactionDTO @param creditRequestDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	addStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	addStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	addStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	This represents the addBillPaymentSchedule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	addStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	This represents the wrapper service for addBillPaymentSchedule method of the entity to be used by dashboard. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	addStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	This represents the updateBillPaymentSchedule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateStandingInstruction	@param billPaymentScheduleDTO @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstructionDetails	@param txnRefNumber @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	closeStandingInstructionForBillSchedule	This method will do a reverse lookup to load the bill payment schedule based on the account and SI instruction @param accountId @param instructionNumber @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	closeStandingInstructionForBillSchedule	This method will do a reverse lookup to load the bill payment schedule based on the account and SI instruction @param accountId @param instructionNumber @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstruction	This method will fetch the SI based on the instruction key as stored in the bill payment schedule @param billPaymentScheduleDTO @param txnContext @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstruction	This method will fetch the SI based on the instruction key as stored in the bill payment schedule @param billPaymentScheduleDTO @param txnContext @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstruction	This method will fetch the SI based on the instruction key as stored in the bill payment schedule @param billPaymentScheduleDTO @param txnContext @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstruction	This method will fetch the SI based on the instruction key as stored in the bill payment schedule @param billPaymentScheduleDTO @param txnContext @return @throws FatalException	This represents the fetchBillPaymentSchedule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BillPaymentScheduleDTO,billPaymentScheduleDTO ,billPaymentScheduleDTO TODO: Explain the billPaymentScheduleDTO above @return {@link com.ofss.fc.app.pc.billpay.dto.schedule.BillPaymentScheduleInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	fetchStandingInstruction	This method will fetch the SI based on the instruction key as stored in the bill payment schedule @param billPaymentScheduleDTO @param txnContext @return @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	This service allows the value date to be incremented by specific days. @fcb.param context @fcb.param type @fcb.param collectionType @fcb.param transactionReferenceNo @fcb.param messageId @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	This service allows calculate the difference between value dates and call the extend float By Days service @fcb.param sessionContext @fcb.param transactionLog @fcb.param messageLogItem @fcb.param valueDate @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	manageReverseFees	@param reversalTransactionDTO @param accountId @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	manageReverseFees	@param reversalTransactionDTO @param accountId @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	logPeriodicFeeAction	<p>This method will log a periodic fee action with casa account. This action will get picked and processed in batch on the actual charging date.</p> @param accountId @param actionDate @param periodicFeeActionType @param earmarkId @param pricingTicketNumber @param transactionReferenceNumber @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	logPeriodicFeeAction	<p>This method will log a periodic fee action with casa account. This action will get picked and processed in batch on the actual charging date.</p> @param accountId @param actionDate @param periodicFeeActionType @param earmarkId @param pricingTicketNumber @param transactionReferenceNumber	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	isBusinessTransferAllowed	<p>This method will fetch the offer id of the account which has initiated the business transfer transaction and validate if such a transaction is allowed for this account, which is specified at the offer level under which this account has been opened. </p> @param accountId @param fundTransferType @return	<p>This method will fetch the offer id of the account which has initiated the business transfer transaction and validate if such a transaction is allowed for this account, which is specified at the offer level under which this account has been opened. </p> @fcb.param MI, String, accountId @fcb.param MI, FundTransferType, fileType <li> {@link com.ofss.fc.enumeration.pc.FundTransferType}</li> @return isBusinessTransferAllowed @throws BusinessException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	isBusinessTransferAllowed	<p>This method will fetch the offer id of the account which has initiated the business transfer transaction and validate if such a transaction is allowed for this account, which is specified at the offer level under which this account has been opened. </p> @param accountId @param fundTransferType @return	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	isBusinessTransferAllowed	<p>This method will fetch the offer id of the account which has initiated the business transfer transaction and validate if such a transaction is allowed for this account, which is specified at the offer level under which this account has been opened. </p> @param accountId @param fundTransferType @return	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCCASAAadapter	checkIfRequiresCFCAccounting	<p>This method will read the interest calculation nature from the account and intimate payments if CFC accounting is required or not. </p> @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	checkIfRequiresCFCAccounting	<p>This method will read the interest calculation nature from the account and intimate payments if CFC accounting is required or not. </p> @param accountId @return	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	reverseMultiFundDeposit	<p>This method initiates the reversal of Multi Fund Deposit transaction with required information populated in the accountTransactionDTOList </p> @param accountTransactionDTOList @throws FatalException	
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	reverseMultiFundDeposit	<p>This method initiates the reversal of Multi Fund Deposit transaction with required information populated in the accountTransactionDTOList </p> @param accountTransactionDTOList @throws FatalException	This method performs the reversal action for the Multi Fund Deposit transaction. @param accountTransactionDTOList @throws FatalException
IPCCASAAadapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCCASAAadapter	reverseMultiFundDeposit	<p>This method initiates the reversal of Multi Fund Deposit transaction with required information populated in the accountTransactionDTOList </p> @param accountTransactionDTOList @throws FatalException	@param sessionContext @param transactionReferenceNo @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceOriginationAdapter	updateLMI PolicyDetails		This method populates the LMIResponseDTO and returns the same
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method would intimate collection on account credit matrix updates/close/re-open @param accountId @param feederDTO
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	The method is used to update collection module when a subsequent disbursement event is triggered on a loan account already in collection. @param appCtx @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param schedule
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called when full writeoff is performed on loan account to update collections data for in collections accounts. @param scheduleId @param writeOfAmount @param writeOffDate
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called when preference update is performed on an in collection account. This method will updateAccount preference information to collection @param appCtx @param accountId
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called when user defined status change is performed on an in collection account. This method will update account status to collection @param appCtx @param loanAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called to update only the unclear balance amount change to collections. Use case is check posting and check dishonour. @param appCtx @param accountId @param unclearAmount
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called during asset classification when only crr code is changed and not accrual status. Will be called to update collection with new CRR code. @param appCtx @param accountId @param crrCode
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Update CRR code and accrual status to collection this is called during asset classification if in collection flag is true. @param appCtx @param accountId @param crrCode
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called when installment payment is received in loan account which is in collections. It will update collections data. @param loanAccount @param transactionDate @param transactionRefNo @param valueDate @param transactionAmount @param accountAmount @param debitCredit @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called when Restructure transaction is performed in loan account which is in collections. It will update collections data. @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Updates collections for Redraw. @param appCtx @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Updates collections for with common account details, arrears and schedule. @param appCtx @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is called for transactions when common account details and arrears are updated in an incollection account. @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Updates collections for Branch Transfer. @param appCtx @param loanAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Updates collection for Offser Swap. @param appCtx @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	Updates Colletion for account closure @param appCtx @param loanAccount
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method updates the common account information to collection. @param appCtx @param feederDTO @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateAccountDetails	This method is called to update the account basic information to collections module. @param accountFeederInformationDTO	This method is responsible for updating the System Status of Loan Account, which gets changed when an Account Closure Request is raised. @param applicationContext @param loanAccount
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateWarningIndicators	This method is called to update the state of a warning indicator to collections module. @param accountId @param warningIndicators	This helper method is called to update warning indicators to collections. @param appCtx @param loanAccount @param warningIndicators
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateRepaymentSchedule	The Repayment schedule for the term loan account is updated to collections module by calling this method. @param accountId @param repaymentSchedule	This method updates the common account information to collection. @param appCtx @param feederDTO @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updateArrearData	Update changes to an existing arrear to Collections Module @param accountId @param arrearDetails	This method updates the common account information to collection. @param appCtx @param feederDTO @param loanAccount @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear @param loanRepaymentScheduleDTOs
ILoanCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanCollectionAdapter	updatePaymentTransactionDetails	This method will be called when any payment is received on an incollection account. This will update collections module about the transaction details like transaction reference number, transaction amount etc. @param accountPaymentTransactionDetails DTO	This method is called when installment payment is received in loan account which is in collections. It will update collections data. @param loanAccount @param transactionDate @param transactionRefNo @param valueDate @param transactionAmount @param accountAmount @param debitCredit @param arrearDetails @param sumofAllArrearsDTO @param principalArrearRaisedDTO @param nonStarter @param noOfInstallmentsInArrear
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchNextWorkingDate	This method will interact with common services to fetch next working date for a given branch depending on the given current working date @param branchId @param currentDate @return @throws FatalException	This method will be used to calculate and update next working date of the branch. @param branchDetails
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchNextWorkingDate	This method will interact with common services to fetch next working date for a given branch depending on the given current working date @param branchId @param currentDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchNextWorkingDate	This method will interact with common services to fetch next working date for a given branch depending on the given current working date @param branchId @param currentDate @return @throws FatalException	@param sessionContext @return @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchDenominationForBranchCurrency	This method will interact with common services to fetch configured denominations for a given currency code @param branchCurrencyDetails @return @throws FatalException	This method will fetch denominations for a given currency by interfacing with Common Service. @param currencyCode @return @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchDenominationTrackingRequiredFlag	This method will interact with common services and fetch if denomination tracking is required at bank level @return @throws FatalException	
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchDenominationTrackingRequiredFlag	This method will interact with common services and fetch if denomination tracking is required at bank level @return @throws FatalException	This method will interact with common services and fetch if denomination tracking is required at bank level @return @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchCashHandlingRequiredFlag	This method will interact with common services and fetch if cash handling is allowed for bank and branch code @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchCashHandlingRequiredFlag	This method will interact with common services and fetch if cash handling is allowed for bank and branch code @return @throws FatalException	This method will interact with common services and fetch if cash handling is allowed for bank and branch code @return @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	isControllingBranch	Returns whether the given branch is the HO (Head Office) or not. @param branchId the branch id. @return <tt>true</tt> if the branch is the HO, <tt>>false</tt> otherwise.	This method is used to fetch values like posting date, next posting date and previous posting date which is required by the UI channels to construct sessionContext @param com.ofss.fc.app.context.SessionContext,sessionContext,sessionContext @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	isControllingBranch	Returns whether the given branch is the HO (Head Office) or not. @param branchId the branch id. @return <tt>true</tt> if the branch is the HO, <tt>>false</tt> otherwise.	
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchReportingBranches	This method will return reporting branches for a given branch. @param branchId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchReportingBranches	This method will return reporting branches for a given branch. @param branchId @return	This represents the fetchBySearchCriteria method of the entity. This method will fetch batch operations for a given search criteria. Batch type is a mandatory parameter. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,com.ofss.fc.app.brop.dto.batchoperations.BatchOperationSearchDTO ,searchCriteriaDTO,,searchCriteriaDTO @return {@link com.ofss.fc.app.brop.dto.batchoperations.BatchOperationInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchReportingBranches	This method will return reporting branches for a given branch. @param branchId @return	If branch id in search criteria is not specified, then details for all the reporting branches of logged in branch also need to be fetched. @param branchId @param controllingBranch @return
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchHoliday	This method will interact with common services to fetch next working date for a given branch depending on the given current working date @param branchId @param date @return @throws FatalException	This method will interact with common services to find if the given date is a holiday for a branch @param branchId <tt>String</tt> @param date <tt>com.ofss.fc.datatype.Date</tt> @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchHoliday	This method will interact with common services to fetch next working date for a given branch depending on the given current working date @param branchId @param date @return @throws FatalException	
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchNthWorkingDay	This method will interact with common services to find working date for the branch as per prior days to raise indent maintained for a branch. @param branchId <tt>String</tt> @param numberOfDays <tt>int</tt> @param currentDate <tt>com.ofss.fc.datatype.Date</tt> @return @throws FatalException	This method will interact with common services to find working date for the branch as per prior days to raise indent maintained for a branch. @param branchId <tt>String</tt> @param numberOfDays <tt>int</tt> @param currentDate <tt>com.ofss.fc.datatype.Date</tt> @return @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonServicesAdapter	fetchNthWorkingDay	This method will interact with common services to find working date for the branch as per prior days to raise indent maintained for a branch. @param branchId <tt>String</tt> @param numberOfDays <tt>int</tt> @param currentDate <tt>com.ofss.fc.datatype.Date</tt> @return @throws FatalException	This service determines if the date been given is a working date and in the range of prior days to raise indent limit maintained for the given branch. @param sessionContext @param indentDate @return @throws com.ofss.fc.infra.exception.FatalException
ILegalProcessesForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ILegalProcessesForApprovalRequestAdapter	initiateLegalProcessForApprovalProcess		@see com.ofss.fc.app.collection.service.legalprocessforapproval.ILegalProcessForApprovalApplicationService This method creates case from LegalProcessForApprovalRequestDTO. @param p_SessionContext @param p_LegalProcessForApprovalRequestDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILegalProcessesForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ILegalProcessesForApprovalRequestAdapter	initiateLegalProcessForApprovalProcess		
IDDALImplementationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\impl\dda\balances\IDDALImplementationAdapter	getOrderedAcquisitionList	This method gives acquisition list in implementation of L as per requirement in product. The order of implementation is   1. TemporaryOverdraft  2. TemporaryExcess  3. Overline  This may vary as per site. So corresponding person can provide its own implementation after overriding this adapter.	This method will derive applicable acquisition buckets for L.Product is going to give 'L' as bucket. And This method will call the factory to get the proper implementation of L. Because implementation of L is site specific. This method will give call to a factory DDALImplementationFactory.getAcquisitionList(). this method implementation is site specific. that method implementation should be in order. Not to be used now.
IDDALImplementationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\impl\dda\balances\IDDALImplementationAdapter	getOrderedApportionList	This method gives apportion list in implementation of L as per requirement in product. The order of implementation is   1. TemporaryOverdraft  2. Overline  3. TemporaryExcess  This may vary as per site. So corresponding person can provide its own implementation after overriding this adapter.	This method will derive applicable apportion buckets for L.Product is going to give 'L' as bucket. And This method will call the factory to get the proper implementation of L. Because implementation of L is site specific. This method will give call to a factory DDALImplementationFactory.getOrderedApportionList(). this method implementation is site specific. that method implementation should be in order. Not to be used now.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerDetails	<p>This method returns customer details on the basis of the customer Id passed. It returns all the information about a customer.</p> <p>@fcb.param  MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;,&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; @param applicationContext  @param customerId @return  CustomerDetailsDTO</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerRelationDetails	<p>This method returns specific customer details on the basis of the customer Id passed. It returns customername, customerid and ownership status</p> <p><b>@fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; <b>@fcb.param MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; <b>@param applicationContext @param customerId @return customerDetailsDTOList</b></p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerRelationDetailForAccount	<p>This method returns specific customer details on the basis of the Account Id passed. It returns customername, customerid and ownership status</p> <p><b>@fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt;</p> <p><b>@fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> A valid account Id for which EarMark and limit has to be checked&lt;br&gt;</p> <p><b>@param applicationContext @param accountId @return customerDetailsDTOList</b></p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerRelationDetailForAccount	<p>This method returns specific customer details on the basis of the Account Id passed. It returns customername, customerid and ownership status</p> <p><b>@fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; <b>@fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> A valid account Id for which EarMark and limit has to be checked&lt;br&gt; <b>@param applicationContext @param accountId @return customerDetailsDTOList</b></p>	If customer relation is other than Sole Owner

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getAccount Holders	<p>This method returns Account Holder details on the basis of the Account Id passed. It returns all the information about a Account Holder.</p> <p><b>@fcb.param</b>  <b>MI,ApplicationContext</b>,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;,&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; <b>@fcb.param</b>  <b>MI,String</b>,&lt;b&gt;&lt;u&gt;AccountId&lt;/b&gt;&lt;/u&gt;&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid Account Id for which EarMark and limit has to be checked&lt;br&gt; <b>@param</b>  <b>applicationContext</b> <b>@param</b>  <b>accountId</b> <b>@return</b> List&lt;Long&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getAccount Holders	<p>This method returns Account Holder details on the basis of the Account Id passed. It returns all the information about a Account Holder. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;AccountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid Account Id for which EarMark and limit has to be checked&lt;br&gt; @param applicationContext @param accountId @return List&lt;Long&gt;</p>	<p>This method is used to link the external account to loan account for principal offset. It stores all the necessary information of the external account linked to the loan account. @param applicationContext @param extOffsetAccountDetails</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerAddressDetails	<p>This method returns the address details of a customer on the basis of the customer Id passed.</p> <p>@fcb.param  MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;,&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; @param  applicationContext @param  customerId @return  CustomerAddressDetailsDTO  @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerBlackListDetails	<p>This method returns the black list details of a customer on the basis of customer Id passed. If the customer is black listed then it will return CustomerBlackListDetailsDTO.</p> <p><b>@fcb.param</b>  <b>MI,ApplicationContext</b>,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/u&gt;&lt;/b&gt;,&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; <b>@fcb.param</b>  <b>MI,String</b>,&lt;b&gt;&lt;u&gt;CustomerId&lt;/u&gt;&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; <b>@param context @param customerId @return</b>  CustomerBlackListDetailsDTO  <b>@throws</b> FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getCustomerMemoDetails	<p>This method returns the customer memos on the basis of customer Id. It returns all memos that are present for a customer. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; @param context @param customerId @return CustomerMemoDetailsDTO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	insertCustomerAccountRelationship	<p>This method inserts a customer and account relationship. Basically, any customer can have multiple accounts, or multiple accounts can have same customer Id.</p> <p><b>@fcb.param</b>  <b>MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;</b>,  <b>&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; <b>@fcb.param</b>  <b>MI,CustomerAccountRelationshipDetailsDTO,&lt;b&gt;&lt;u&gt;relationshipDetailsDTO&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt;</b> Complete DTO carrying complete information about customer and &lt;br&gt; account relationship with all required details.&lt;b&gt; @param applicationContext @param relationshipDetailsDTO</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	insertCustomerAccountRelationship	<p>This method inserts a customer and account relationship. Basically, any customer can have multiple accounts, or multiple accounts can have same customer Id.</p> <p>@fcb.param  MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param  MI,CustomerAccountRelationshipDetailsDTO,&lt;b&gt;&lt;u&gt;relationshipDetailsDTO&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; Complete DTO carrying complete information about customer and &lt;br&gt; account relationship with all required details.&lt;b&gt; @param  applicationContext @param  relationshipDetailsDTO</p>	<p>This method performs updates to tables in CIF domain during the account opening process. @fcb.param  MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; &gt; A valid customer Id for which EarMark and limit has to be checked&lt;br&gt; @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid accountId @fcb.param  MI,int,&lt;b&gt;&lt;u&gt;productCode&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid product Code @fcb.param  MI,int,&lt;b&gt;&lt;u&gt;branchCode&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid branch Code @fcb.param  MI,RelationType,&lt;b&gt;&lt;u&gt;customerRelation&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid customer Relation @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;makerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid makerId @fcb.param  MI,String,&lt;b&gt;&lt;u&gt;checkerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid checkerId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	performAccountOpeningCIFUpdates	<p>This method is called during Loan Account Opening. It performs CIF Updates, it inserts in the basic tables of CIF: 1-&gt; ci_cust_acct_rest_detls 2-&gt; SM_ACCESS_CODE_CUST_XREF 3-&gt; cs_ho_custacctxref 4-&gt; ch_acct_cust_xref based on following parameters. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;CustomerId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid customer Id for which EarMark and limit has to be checked.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;AccountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid Account No for Loan Account Opening has to be passed.&lt;br&gt; @fcb.param MI,int,&lt;b&gt;&lt;u&gt;ProductCode&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid Product Code for Loan Account Opening has to be passed.&lt;br&gt; @fcb.param MI,int,&lt;b&gt;&lt;u&gt;BranchCode&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid Branch Code for Loan Account Opening has to be passed.&lt;br&gt; @fcb.param</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	createPartyAccountRelation	This method is used to store the details of all the parties @param partyAccRelationshipAdapterDto @throws FatalException	This method is used to call the service PartyAccountRelationshipApplicationService.changePartyAccountRelationship(SessionContext, PartyAccountRelationshipDTO) which is used to store the details of all the parties at the time of account Opening. @param partyAccRelationshipAdapterDto
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getLegalNameOfParty	This method fetches the name of the party, from a given partyId. @param context @param partyId @return String @throws FatalException	
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getLegalNameOfParty	This method fetches the name of the party, from a given partyId. @param context @param partyId @return String @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	<p>This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;App            licationContext&lt;/b&gt;&lt;/u&gt;,&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid account Id For which relation is already maintained&lt;br&gt; @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException</p>	<p>This method will populate all the fields that are needed to open a new loan account from SplitLoanAccountInstructions. The following features of loan account will be copied to new loan account from original loan account. &lt;li&gt;Customer to Account Relationship&lt;/li&gt; &lt;li&gt;Loan Account Payment Instructions&lt;/li&gt; &lt;li&gt; Loan Account Attributes&lt;/li&gt; &lt;li&gt;Redraw Allowed Feature&lt;/li&gt; @param splitAccountInstructions</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	<p>This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid account Id For which relation is already maintained&lt;br&gt; @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException</p>	<p>This method will populate all the fields that are needed to open a new loan account from SplitLoanAccountInstructions. The following features of loan account will be copied to new loan account from original loan account. &lt;li&gt;Customer to Account Relationship&lt;/li&gt; &lt;li&gt;Loan Account Payment Instructions&lt;/li&gt; &lt;li&gt; Loan Account Attributes&lt;/li&gt; &lt;li&gt;Redraw Allowed Feature&lt;/li&gt; @param splitAccountInstructions</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	<p>This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;App            licationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/b&gt;&lt;br&gt; A valid account Id For which relation is already maintained&lt;br&gt; @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException</p>	<p>This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allwoed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>, <b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</b></u><b>:</b></u><b>:</b>  A valid account Id For which relation is already maintained  @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException	This method returns specific customer details on the basis of the Account Id passed. It returns customername, customerid and ownership status @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>, <b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</b></u><b>:</b>  A valid account Id for which EarMark and limit has to be checked  @param applicationContext @param accountId @return CustomerRelationshipDetailsDTOList



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	<p>This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;accountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid account Id For which relation is already maintained&lt;br&gt; @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException</p>	<p>This method returns Account Holder details on the basis of the Account Id passed. It returns all the information about a Account Holder. @fcb.param MI,ApplicationContext,&lt;b&gt;&lt;u&gt;ApplicationContext&lt;/b&gt;&lt;/u&gt;, &lt;b&gt;:&lt;/b&gt;&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. After &lt;br&gt; successful authentication the user request is processed.&lt;br&gt; @fcb.param MI,String,&lt;b&gt;&lt;u&gt;AccountId&lt;/b&gt;&lt;/u&gt;&lt;b&gt;:&lt;/b&gt;&lt;br&gt; A valid Account Id for which EarMark and limit has to be checked&lt;br&gt; @param applicationContext @param accountId @return List&lt;Long&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,<b><u>App</u>licationContext</b></u>,<b>:</b></b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</u></b></u><b>:</b></b>  A valid account Id For which relation is already maintained  @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException	This method returns all the aprty to account retaiions details for the inputed account. @fcb.param MI,ApplicationContext,<b><u>App</u>licationContext</b></u>,<b>:</b></b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</u></b></u><b>:</b></b>  A valid account Id For which relation is already maintained  @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllPartiesOfAccount	This method returns all the aprty to account retentions details for the inputed account. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>, <b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</b></u><b>:</b>  A valid account Id For which relation is already maintained  @param applicationContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException	This method returns primary Account Holder id on the basis of the Account Id passed. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>, <b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>AccountId</b></u><b>:</b>  A valid Account Id for which EarMark and limit has to be checked  @param applicationContext @param accountId @return List<Long>
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchAllAccountsOfPartySet		<pre>This method is used to fetch the Transaction Accounts . &lt;br&gt; The following steps are carried out in this method:- &lt;br&gt; 1) Call is given to ILoanPartyInformationApplicationAdapter.fetchAllAccountsOfPartySet to fetch all the transaction account number of Loan's party holder/s 2) Call is given to ILoanDDAadapter.inquireDDAAccount to fetch the details of the Transaction Accounts. </pre> @param sessionContext @param loanAccountId @throws FatalException @returns BasicInquiryDetailsDTO[]

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method returns specific customer details on the basis of the Account Id passed. It returns customername, customerid and ownership status @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</u></b></u>, ,<b>:</b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After   successful authentication the user request is processed.  @fcb.param MI,String,<b><u>accountId</u></b></u><b>:</b>  A valid account Id for which EarMark and limit has to be checked  @param applicationContext @param accountId @return CustomerRelationshipDetailsDTOList
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getPartyDetails	This method is used to get the partydetails based on the partyId @param partyId @return	
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not. @param partyId	The method checks if the customer is blacklisted. Internally, the method calls party adapter and checks if the party is blacklisted. Refer {@link ILoanPartyInformationApplicationAdapter#isCustomerBlacklisted}
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	isCustomerBlacklisted	This method is used to check whether the customer is blackListed or not. @param partyId	
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchCustomerMemo	This method will return all the memos attached to the party associated with the loan account. @param partyId @return	The method checks if the customer has a memo. Internally the method connects with the party adapter and checks if the customer has a memo. Refer {@link IDDAPartyAdapter#fetchCustomerMemo}
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchCustomerMemo	This method will return all the memos attached to the party associated with the loan account. @param partyId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	getPrimaryAccountHolder	This method returns primary Account Holder id on the basis of the Account Id passed. @fcb.param MI,ApplicationContext,<b><u>ApplicationContext</b></u>,<b>:</b></b>  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. After  successful authentication the user request is processed. @fcb.param MI,String,<b><u>AccountId</b></u><b>:</b></b>  A valid Account Id for which EarMark and limit has to be checked  @param applicationContext @param accountId @return List<Long>	This method triggers alert for updating notifications. @param applicationContext @param partyId @param source
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchWarningIndicatorOnParty		This method validates the party warning indicator. @param sessionContext @param account
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchWarningIndicatorOnParty		This method validates the party warning indicator present at party level. This will not allow to do the setup for principal offset in ase party has warning indicator type as "PREVENT_PRINCIPAL_OFFSETS". @param sessionContext @param account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchWarningIndicatorOnParty		<pre> This method checks if Repayment review is required or not it return true if Repayment review is required otherwise false. it checks the following conditions. &lt;li&gt;TRUE: If Rate change action is present, otherwise False.&lt;/li&gt; &lt;li&gt;False: If Repayment Restriction has been maintained for the account, otherwise TRUE.&lt;/li&gt; &lt;li&gt;False: If warning Indicator like 'Notice of Bankruptcy' is present, otherwise TRUE.&lt;/li&gt; </pre> <pre> @param applicationContext @param accountId @param processDate @param nextProcessDate @return isRepaymentReviewIsRequired boolean </pre>
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	fetchWarningIndicatorOnParty		<p>This method is responsible for checking the presence of Warning Indicators for Loan Account holder. If the Warning Indicators have been maintained for the Party, the method throws error. @param sessionContext @param loanAccount @throws FatalException</p>
ILoanPartyInformationApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPartyInformationApplicationAdapter	checkIfWarningIndicatorsDefinedForParty		<p>This method is used to check warning indicator at party level. @param partyToAccountRelationshipDTO</p>
IBatchReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\batch\IBatchReportAdapter	genBatchReport		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBatchReport Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\batch\IBatchReportAdapter	genBatchReport		(non-Javadoc) @see com.ofss.fc.bh.batch.ShellRootHelper#processBatchRequest(com.ofss.fc.infra.xml.XMLParser) This method will initiate the report generation. Algorithm <ol><li>Get the {@link #ReportType} for the batch category by calling method {@link getReportType()}. <li>Call {@link ReportJobRequestor#genBatchReport(String, String, String, String, String, String)} to initiate the report generation. </ol>
IBatchReport Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\batch\IBatchReportAdapter	processPendingReports		
IBatchReport Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\batch\IBatchReportAdapter	processPendingReports		(non-Javadoc) @see com.ofss.fc.bh.batch.ShellRootHelper#processBatchRequest(com.ofss.fc.infra.xml.XMLParser) This method will initiate the report generation. Algorithm <ol><li>Get the {@link #ReportType} for the batch category by calling method {@link getReportType()}. <li>Call {@link ReportJobRequestor#genAbortedBatchReport(String, String, String, String, String, String)} to initiate the report generation. </ol>
IntegrationAnalyticsOPS Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\analytics\IntegrationAnalyticsOPSAdapter	startOrRestartProcess		Starts or restarts the analytics Category @param sessionContext @param eodCommandType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationAnalyticsOPSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\analytics\IntegrationAnalyticsOPSAdapter	startOrRestartProcess		
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method will be called in case no record is found in account address linkage table when queried from ACCT020Account preference Screen. In such scenario, all the parties and their corresponding party details for the given account number are fetched. Next the associated parties and their party details are fetched for the account and PartyAddressLinkageDTO is populated with the same. AdditionalPartyAddressLinkageDTOs is populated with the party which are linked to the account and have relationship other than Others. @param sessionContext @param accountAddressLinkageInquiryDTO @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method can be used during create mode to fetch all the parties that are linked to an account and also the basic details of the party. @param sessionContext @param accountAddressLinkageInquiryDTO @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method will fill the PartyAddressLinkageDTO with all the required details. @param sessionContext @param accountId @param partyAddressLinkageDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object {@link TitleModificationResponseDTO}. Add the array of account ids by calling toArray() on the arraylist. &lt;li&gt;Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException</li> </ul> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method is used for preparing the Operating instructions signatories DTO which will be used for consumption by the SMS module for sending the transactions to authorization in case of channels like IB,MB. <b>Since:</b> R2.2 @param operatingInstructionPartyAccountDTO @param operatingInstructionAuthorizedSignatoriesDTO @param operatingInstructionDetailsDTOs @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method is used to fetch the List of parties from the List of OperatingInstructionPartyDetailsDTO @param operatingInstructionsPartyDetailsList @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	<p> For the input account id, this method will fetch the party name and the ownership type. </p> @param accountId @param partyNameBuffer @param accountOwnerTypeBuffer @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method will populate bundle owner details. @fcb.param MI, PartyBundleDetailsDTO[], partyBundleDetailsDTO, Indicates the bundle details held by the input party. @fcb.param MI, int, index, Indicates the index at which to set values. @fcb.param MI, Set<String>, bundleOwnerSet, Indicates the list of bundle owners. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	@param sessionContext @param primaryHolderPartyId Primary account holder of the account @param partyMap Map with all parties in the account as keys @param validPartyMap Map with all valid party types as key @param accountNomineeDTO @throws BusinessException @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyDetails	@param partyId @return @throws FatalException	This method fetches party names for all the parties present in the input. @param sessionContext @param nominationStructure @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method validates the DDA account no specified in the settlement, checks whether the DDA account holder and the Loan Account holders are the same. @param componentPreferenceDTO @param partyToAccountRelationshipDTOs @param amendEvent @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method fetches the account type It can be either CASA or TD @param MI,String,accountId, , Indicates the account Id. @return Returns the account type. @throws com.ofss.fc.infra.exception.FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	@param applicationContext @param accountId @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6) Create an instance of the Domain Service

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>(AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method will be called in case no record is found in account address linkage table when queried from ACCT020Account preference Screen. In such scenario, all the parties and their corresponding party details for the given account number are fetched. Next the associated parties and their party details are fetched for the account and PartyAddressLinkageDTO is populated with the same. AdditionalPartyAddressLinkageDTOs is populated with the party which are linked to the account and have relationship other than Others. @param sessionContext @param accountAddressLinkageInquiryDTO @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method can be used during create mode to fetch all the parties that are linked to an account and also the basic details of the party. @param sessionContext @param accountAddressLinkageInquiryDTO @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method will fill the PartyAddressLinkageDTO with all the required details. @param sessionContext @param accountId @param partyAddressLinkageDTO @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method calls offsetAccountBusinessPolicy @param sessionContext @param offsetAccountDTO @return @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	<p>This method will validate business policies for addition or deletion of bundle owner(s) and if the deletion of any bundle owner(s) is causing delinkage of that owner's account from the bundle, this method will populate the list of accounts that will be delinked and return the same. </p> @param partyBundleLinkage @param updatedPartyIds @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	<p>This method will check if the party of the fetched account is a bundle owner. </p> @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, String, accountParty, Indicates the account holder. @fcb.param MI, String, accountId, Indicates the account number. @fcb.param MI, String, accountOfferId, Indicates the offer to which the account belongs. @fcb.param MI, String, bundleId, Indicates the bundle chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscriptionDetailsDTOArray, Indicates the subscription details. @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	The following method is used to fetch account related information. The following method is used to avoid Interaction begin and close related operations. @param accountId @param accountType @return @throws FCRException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	<p> For the input account id, this method will fetch the party name and the ownership type. </p> @param accountId @param partyNameBuffer @param accountOwnerTypeBuffer @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method will fetch the warning indicators applied for given event and account code
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method tags the bundle qualification status to the offers.  By default all the offer will be marked with bundle qualification as APPLIED or APPLIED/EXISTING depending on whether the offer has been applied and if there is also an existing account under the offer.  If the input offers are already linked to the un-associated account then offers will be marked as Existing.  If the input offers are not linked to the un-associated account then offers will be marked as APPLY @param bundleOfferDTOArray @param bundleOwnerAccountDetailsDTOArray @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				subscribedOfferIdList @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesofAccount	@param accountId @return @throws FatalException	This method fetches the account type It can be either CASA or TD @fcb.param MI,String,accountId, , Indicates the account Id. @return Returns the account type. @throws com.ofss.fc.infra.exception.FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkGuardianDetails	@param partyId @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkGuardianDetails	@param partyId @throws FatalException	@param sessionContext @param primaryHolderPartyId Primary account holder of the account @param partyMap Map with all parties in the account as keys @param validPartyMap Map with all valid party types as key @param accountNomineeDTO @throws BusinessException @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	This method will fetch All linked account details for party Id. @param sessionContext @param partyId @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsOfParty	This method fetches all Accounts of a party @param sessionContext @param partyId @throws FatalException	This method fetches the transactions made in all accounts linked to the partyId in the PartyTransactionInquiryDTO @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext ,This parameter captures the basic information about the user of the system.  The information is used by the system for authentication. @fcb.param MI,com.ofss.fc.app.account.dto.statement.PartyTransactionInquiryDTO,partyTransactionInquiryDTO ,This parameter is the PartyTransactionInquiryDTO which contains the partyId and StatementInquiryFilterInputDTO (contains the filter conditions) @fcb.param MI,com.ofss.fc.app.search.dto.PageInformationDTO,pageInformationDTO , , Contains the information for which page needs to be retrieved. @return {@link com.ofss.fc.app.account.dto.statement.PartyTransactionInquiryResponse} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkIfPartyRelatedToAccount	This method will check whether the party is linked to account or not. @param sessionContext @param accountId @param partyId @throws FatalException	This method will check whether the party is linked to account or not. @param sessionContext @param accountId @param partyId @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkIfPartyRelatedToAccount	This method will check whether the party is linked to account or not. @param sessionContext @param accountId @param partyId @throws FatalException	This method will fetch all the addresses of the party. @param sessionContext @param partyId @return PartyAddressDTO[]
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkIfPartyRelatedToAccount	This method will check whether the party is linked to account or not. @param sessionContext @param accountId @param partyId @throws FatalException	This method will validate all the partyIds that have been linked to account address @param sessionContext @param accountAddressLinkageDTO
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAddressesOfParty	This method will fetch all the addresses of the party. @param sessionContext @param partyId @return PartyAddressDTO[]	This method will fetch all the addresses of the party. @param sessionContext @param partyId @return PartyAddressDTO[]
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method will return the postal address of the party depending upon the address type choosed. @param sessionContext @param partyId @param addressType @return PostalAddress
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	Sets the postal address details.Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method fetches the address linkages for a particular party associated with the account. In the Inquiry DTO the docTypeId field would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiry Response. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the inputs. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6) Check for the validity of the party and its account relationship by calling fetchPartyAccountRelationship() method exposed by the PartyAccountRelationshipApplicationService with accountId and partyId as the input. If the party is not valid, "Invalid Party" error message is displayed. If the party does not have any relationship with the account, "No Relationship with the Account" error message is displayed. 7) Create an instance of the Domain Service (AccountAddressLinkageService). 8) Call the fetchPartyAddressLinkage() method of the AccountAddressLinkageService with accountId and

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>partyId as the input. 9) Receive the AccountAddressLinkage entity in response. 10) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) Call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) Call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO @param sessionContext</p>
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	<p>This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress</p>	<p>This method will fetch the address related to party and the address type. @param sessionContext @param accountId @param partyId @param postalAddressType @return DocumentAddressLinkageInquiryResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method will fetch the postal address of the party for the current address type. @param sessionContext @param accountId @param partyId @param addressType @return PostalAddress
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method will fetch the postal address of the party for the current address type. @param sessionContext @param accountId @param partyId @param addressType @return PostalAddress
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method will check for default communication address as it is a mandatory field. @param accountAddressLinkage
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyAddress	This method will return the postal address of the party depending upon the address type chosen. @param sessionContext @param partyId @param addressType @return PostalAddress	This method will check for default communication address of all the parties linked to account.  <b>Since:</b> R2.2   @param accountAddressLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountBundleOfDetails	This method will return the offer id under which the supplied account has been opened. @fcb.param MI, String, accountId, Indicates the account id for which basic details have been requested for. @return {@link com.ofss.fc.app.adapter.account.offerbundling.dto.AccountBasicDetailsDTO}	@see com.ofss.fc.app.dda.service.core.IDemandDepositInstructionService#createSweepInInstruction(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String)
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountBundleOfDetails	This method will return the offer id under which the supplied account has been opened. @fcb.param MI, String, accountId, Indicates the account id for which basic details have been requested for. @return {@link com.ofss.fc.app.adapter.account.offerbundling.dto.AccountBasicDetailsDTO}	@see com.ofss.fc.app.dda.service.core.IDemandDepositInstructionService#createSweepInInstruction(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String)
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountBundleOfDetails	This method will return the offer id under which the supplied account has been opened. @fcb.param MI, String, accountId, Indicates the account id for which basic details have been requested for. @return {@link com.ofss.fc.app.adapter.account.offerbundling.dto.AccountBasicDetailsDTO}	This method will fetch the details of each supplied account id.  @fcb.param MI, AccountDetailsDTO[], accountDetailsDTOArray, Indicates the list of account ids to be linked to the bundle at the time of bundle subscription. @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountBundleOfferDetails	This method will return the offer id under which the supplied account has been opened. @fcb.param MI, String, accountId, Indicates the account id for which basic details have been requested for. @return {@link com.ofss.fc.app.adapter.account.offerbundling.dto.AccountBasicDetailsDTO}	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p>&lt;p&gt; Method does maintenance fee processing for account modules &lt;/p&gt;</p> <p>&lt;p&gt; It fetches the account basic details having information of party, currency etc. &lt;/p&gt;</p> <p>&lt;p&gt; For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data &lt;/p&gt;</p> <p>@param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p> Method does maintenance fee processing for account modules </p> <p> It fetches the account basic details having information of party, currency etc. </p> <p> For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data </p> @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p> If the new nominated account is a CASA account, this method will do CASA specific validations and update the nominated account. </p> @param partyBundleLinkage @param componentPreferenceDTO @param newBundleAccountLinkage
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p> If the new nominated account is a LOANS account, this method will do CASA specific validations and update the nominated account. </p> @param partyBundleLinkage @param componentPreferenceDTO @param fetchedBundleAccountLinkage
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p>This method will fetch the details of each supplied account id.  </p> @param bundleSubscriptionDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p> During link and delink request, this method will set the currency code of the party bundle linkage to the currency code of the first account in this linkage which is not nominated. </p> @param newBundleAccountLinkageArray @param partyBundleLinkage @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p>This method will validate business policies for addition or deletion of bundle owner(s) and if the deletion of any bundle owner(s) is causing delinkage of that owner's account from the bundle, this method will populate the list of accounts that will be delinked and return the same. </p> @param partyBundleLinkage @param updatedPartyIds @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	<p>This method is exposed to external modules to do fees management for CASA, Loan and settlement module.  Based on the input, the fees will be debited and then intimated to pricing.   In case fee details are not passed as input, then fees will be calculated, debited and intimated to pricing.  Following are the required steps:  <ol> <li>Fetch pricing details   <li>Debit the fee from account.  <li>Raise accounting for the charged fees  <li>Intimate pricing   </ol> <ol> <li>Fetch pricing details If price ticket number is provided in input than inquire fees based on price ticket number. Price ticket number in input indicates that fees is already inquired by the caller. There could be negotiation performed. The final information is stored into the pricing against this price ticket number. else if pricing details are not provided then inquire

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>fees by the event id. It is not know at this point that whether there is fee linked with this event or not. This call to pricing will return PD with amount, NSFTreatment, discounts and additional fee if there is any. If there is no fee then there will not be any further processing. &lt;br&gt; &lt;li&gt;Debit the fee from account. Debit account can be any module account like CASA, Loan, 2. This method will be responsible to debit the fee from account. &lt;br&gt; &lt;li&gt;Raise accounting for the charged fees&lt;br&gt; &lt;li&gt;Intimate the pricing system by calling AccountPricingAdapter.intimatePricing(priceTicketNumber,transactionReferenceNumber): boolean &lt;/ol&gt; &lt;/p&gt; @throws FatalException @fcb.param I,String,priceTicketCode,40,, Dialogue ticket obtained from Pricing for fees applicable. If this is supplied, then wrappedPriceFeatureDTO is not required @fcb.param I,WrappedPriceFeatureDTO,wrappedPriceFeatureDTO,,,Negotiated fees. If this is supplied, then priceTicketCode is not required @fcb.param I,String,arrangementId,40, , Indicates the entity for which fees are applicable during the ongoing event. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI, String, accountCode,40 ,, Indicates the account from which the fees have to be funded @fcb.param I, String, eventId, 40,, Indicates the ongoing event for which the fees are applicable on the entity. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,transactionDesc,255, , Transaction description to be logged on the funding account</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	@param accountCode @param taskModuleType @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchSpecificAccountDetails	Based on the account type obtained from party account relationship, this method makes a call to the corresponding account module to fetch basic details of the account. @param accountId @param accountType @return @throws FatalException	This method will fetch all the fees for the given ticket number. For the fees that are marked as collected, it will raise an accounting event to refund the amount in suspense GL @param applicationContext @param pricingTicketNo @param currency @param domainCategory @param event @param arrangementNo @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRelationForPartyAndAccount	This method takes the party id and the account id as input and fetches the party account relationship. @param partyId @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRelationForPartyAndAccount	This method takes the party id and the account id as input and fetches the party account relationship. @param partyId @param accountId @return @throws FatalException	<p>This method would prepare the OperatingInstructionPartyDetailsDTO.It works as follows: ;For each  OperatingInstructionPartyDetailsDTO, if the operating instruction channel is Branch: Check the Branch signatory  flag set in party account relationship. Check if the KYC for the party is confirmed. Check if the party has valid   user access role(template/account role) defined in OES for the given service. If all the above is true,the party  would be added to list of parties who can operate on the account.If the operating  instruction channel is Internet: Check the Internet signatory flag set in party account relationship Check if the   KYC for the party is confirmed Check if the party has valid user access role(template/account role) defined in  OES for the given service. If all the above is true,the party would be added to list of parties who can operate  on the account.If the operating instruction channel is Mobile Banking: Check the Mobile  Banking signatory flag set in party account relationship Check if the KYC for the party is confirmed Check if the   party has valid user access role(template/account role) defined in OES for the given service. If all the above is   true,the party would be added to list of parties who can operate on the account.; </p> @param sessionContext @param returnDTO @param operatingInstructionPartyDetailsDTOs @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRelationForPartyAndAccount	This method takes the party id and the account id as input and fetches the party account relationship. @param partyId @param accountId @return @throws FatalException	This method would set the party details like the template role, party name, relation to the account. @param accountId @param operatingInstructionPartyDetailsDTO
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRelationForPartyAndAccount	This method takes the party id and the account id as input and fetches the party account relationship. @param partyId @param accountId @return @throws FatalException	This method will return true if account is joint account. @param depositAccount @param hasJointApplicant @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsForParties	This method will fetch all accounts where the input list of parties are the owners. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @return {@link com.ofss.fc.app.party.dto.relation.account.AllPartyAccountRelationshipInquiryResponse} @throws FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsForParties	This method will fetch all accounts where the input list of parties are the owners. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @return {@link com.ofss.fc.app.party.dto.relation.account.AllPartyAccountRelationshipInquiryResponse} @throws FatalException	@see com.ofss.fc.app.dda.service.core.IDemandDepositInquiryService#fetchAccountsOfParty(com.ofss.fc.app.context.SessionContext, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsForParties	This method will fetch all accounts where the input list of parties are the owners. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @return {@link com.ofss.fc.app.party.dto.relationship.account.AllPartyAccountRelationshipInquiryResponse} @throws FatalException	<p>This method will populate party bundle details based on the input party id for the input bundle and subscription details, if supplied. If subscription details are not supplied, it is assumed that it is an existing subscription and the corresponding party bundle linkage will be loaded. </p> @fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of supplied bundle owners. @return {@link PartyBundleDetailsDTO} @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllAccountsForParties	This method will fetch all accounts where the input list of parties are the owners. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @return {@link com.ofss.fc.app.party.dto.relationship.account.AllPartyAccountRelationshipInquiryResponse} @throws FatalException	This method will populate party bundle details based on the input party id for the input bundle and subscription details, if supplied. If subscription details are not supplied, it is assumed that it is an existing subscription and the corresponding party bundle linkage will be loaded. @param partyIdArray @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	validatePartyUpdateFromBundle	If the account to which the party is being modified belongs to a bundle, this method will check if the party being added to the account is a bundle owner. If yes, it will allow the party edition, else it will throw an error to either first add party as bundle owner or delink account from the bundle.  In case a party is being deleted from an account, this method will not do anything. @param partyId @param accountId @param accountModule @param isDelete	If the account to which the party is being modified belongs to a bundle, this method will check if the party being added to the account is a bundle owner. If yes, it will allow the party edition, else it will throw an error to either first add party as bundle owner or delink account from the bundle. In case a party is being deleted from an account, this method will not do anything @param dto @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchVeryBasicPartyDetails	This method will call fetchVeryBasicPartyDetails() in com.ofss.fc.app.party.service.relation.account.PartyAccountRelationshipApplicationService @param accountId @return PartyAccountRelationshipInquiryResponse	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchVeryBasicPartyDetails	This method will call fetchVeryBasicPartyDetails() in com.ofss.fc.app.party.service.relation.account.PartyAccountRelationshipApplicationService @param accountId @return PartyAccountRelationshipInquiryResponse	This method will call fetchVeryBasicPartyDetails() in com.ofss.fc.app.party.service.relation.account.PartyAccountRelationshipApplicationService @param accountId @return PartyToAccountRelationshipInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchVeryBasicPartyDetails	This method will call fetchVeryBasicPartyDetails() in com.ofss.fc.app.party.service.relation.account.PartyAccountRelationshipApplicationService @param accountId @return PartyAccountRelationshipInquiryResponse	This method calls the appropriate transactionInquiry method depending on the type of account. The following are 3 methods called- <ul><li>inquireDDTransactions</li><li>inquireTDTransactions</li><li>inquireLoanTransactions</li></ul> @param sessionContext @param transactionInquiryDTO @param accountType @return StatementItemDTO[] @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchCustomerMemo		
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchCustomerMemo		The following method is used to fetch account memo related details.The following method is used to avoid interaction open and close related operations. @param accountId @param response @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	inquireAll		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	inquireAll		This method fetches the transactions made in all accounts linked to the partyId in the PartyTransactionInquiryDTO @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, ,This parameter captures the basic information about the user of the system.  The information is used by the system for authentication. @fcb.param MI,com.ofss.fc.app.account.dto.statement.PartyTransactionInquiryDTO,partyTransactionInquiryDTO, ,This parameter is the PartyTransactionInquiryDTO which contains the partyId and StatementInquiryFilterInputDTO (contains the filter conditions) @fcb.param MI,com.ofss.fc.app.search.dto.PageInformationDTO,pageInformationDTO , , Contains the information for which page needs to be retrieved. @return {@link com.ofss.fc.app.account.dto.statement.PartyTransactionInquiryResponse} @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		This method would set all the parties in case there is no operating instruction defined for the account. @param sessionContext @param accountId @param returnDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		<pre> This method is used to provide the non financial operating instruction details. When a non-financial transaction being performed on an account is to be processed, Operation executive checks whether any operating instructions are being defined for the respective account or not.This method will accept account Id, transaction Id and party Id as input arguments. The instruction description maintained against the Non-financial operating instruction type would be used in displaying in the text fields for non-financial operating instruction. Following are the steps to be followed in this method:- Create an instance of Inquiry Response Create an object of Assembler Invoke its toDomainObject() method and convert the NonFinancialInstructionRequestDTO to the domain object NonFinancialInstructionRequest. Create an instance of Domain Service and invoke it's fetch method by passing FinancialInstructionRequest. This would return domain object. This domain object will have the instruction details as a response. Invoke the fromDomainObject() method of Assembler and pass the domain object to it. It would return the DTO in response. Set the DTO as a part of InquiryResponse; &amp;Fill the Transaction Status by calling the method fillTransactionStatus and pass the transactionstatus as an argument to it. Set the response status as transaction status </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		<p>This method is used to provide the non financial operating instruction details. When a inquiry based transaction being performed on an account is to be processed, Operation executive checks whether any operating instructions are being defined for the respective account or not. This method will accept account Id, transaction Id and party Id as input arguments. The instruction description maintained against the Inquiry operating instruction type would be used in displaying in the text fields for inquiry operating instruction. Following are the steps to be followed in this method:- Create an instance of InquiryResponse Create an object of Assembler Invoke its toDomainObject() method and convert the InquiryInstructionRequestDTO to the domain object InquiryInstructionRequest. Create an instance of Domain Service and invoke it's fetch method by passing FinancialInstructionRequest. This would return domain object. This domain object will have the instruction details as a response. Invoke the fromDomainObject() method of Assembler and pass the domain object to it. It would return the DTO in response. Set the DTO as a part of InquiryResponse @param sessionContext @param nonFinancialInstructionRequestDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		<p>&lt;p&gt;This method would fetch all the parties related to the account with the following conditions: 1.SOW (Sole owner)&lt;br /&gt; /JAF(Joint and First) /JAO(Joint and Others) in the account. 2.Minors (if Signatory In branch flag of Bank policy &lt;br /&gt; is ticked as per party to party and party to account relationship, document). 3.Party affiliates who have&lt;br /&gt; relationship class as owner, have Signatory In branch flag of Bank policy ticked as per party to party and party&lt;br /&gt; to account relationship, document and are linked to account. 4.Party affiliates who have relationship class as&lt;br /&gt; administrator , have Signatory In branch flag of Bank policy ticked as per party to party and party to account&lt;br /&gt; relationship, document and are linked to account. 5.This method also checks that if the party's KYC is not&lt;br /&gt; completed, the party will not be available in list of required party for operating instruction.&lt;br /&gt; &lt;/p&gt; <b>Since:</b> R2.2 &lt;br&gt;           @param sessionContext @param account @throws FatalException         </p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		<p>&lt;p&gt; This is wrapper method which will fetch all the parties related to the account with the following conditions: 1.SOW (Sole owner)&lt;br /&gt; /JAF(Joint and First) /JAO(Joint and Others) in the account. 2.Minors (if Signatory In branch flag of Bank policy &lt;br /&gt; is ticked as per party to party and party to account relationship, document). 3.Party affiliates who have&lt;br /&gt; relationship class as owner, have Signatory In branch flag of Bank policy ticked as per party to party and party&lt;br /&gt; to account relationship, document and are linked to account. 4.Party affiliates who have relationship class as&lt;br /&gt; administrator , have Signatory In branch flag of Bank policy ticked as per party to party and party to account&lt;br /&gt; relationship, document and are linked to account. 5.This method also checks that if the party's KYC is not&lt;br /&gt; completed, the party will not be available in list of required party for operating instruction.&lt;br /&gt; &lt;/p&gt; &lt;b&gt;Since:&lt;/b&gt; R2.2 &lt;br&gt; @param sessionContext @param account @throws FatalException</p>
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllPartiesOfAccountForOperatingInstruction		<p>@param sessionContext @param operatingInstructionDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	isKYCConfirmed	This method would check if the KYC (Know your customer) for the customer is confirmed @param partyId @return @throws FatalException	This method is responsible for calling the party module inorder to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	isKYCConfirmed	This method would check if the KYC (Know your customer) for the customer is confirmed @param partyId @return @throws FatalException	This method would set the account id,party id,party name,party relation for a given party which will be added to the list of parties who can operate on the account. @param operatingInstructionPartyAccountRelationshipDetails DTO @param count @param partyAccountRelationshipDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	isKYCConfirmed	This method would check if the KYC (Know your customer) for the customer is confirmed @param partyId @return @throws FatalException	<p>&lt;p&gt;This method would prepare the OperatingInstructionPartyDetailsDTO.It works as follows: ;For each&lt;br /&gt; OperatingInstructionPartyDetailsDTO, if the operating instruction channel is Branch: Check the Branch signatory&lt;br /&gt; flag set in party account relationship. Check if the KYC for the party is confirmed. Check if the party has valid &lt;br /&gt; user access role(template/account role) defined in OES for the given service. If all the above is true,the party&lt;br /&gt; would be added to list of parties who can operate on the account.If the operating&lt;br /&gt; instruction channel is Internet: Check the Internet signatory flag set in party account relationship Check if the &lt;br /&gt; KYC for the party is confirmed Check if the party has valid user access role(template/account role) defined in&lt;br /&gt; OES for the given service. If all the above is true,the party would be added to list of parties who can operate&lt;br /&gt; on the account.If the operating instruction channel is Mobile Banking: Check the Mobile&lt;br /&gt; Banking signatory flag set in party account relationship Check if the KYC for the party is confirmed Check if the &lt;br /&gt; party has valid user access role(template/account role) defined in OES for the given service. If all the above is &lt;br /&gt; true,the party would be added to list of parties who can operate on the account.; &lt;/p&gt; @param sessionContext @param returnDTO @param operatingInstructionPartyDetailsDTOs @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	isKYCConfirmed	This method would check if the KYC (Know your customer) for the customer is confirmed @param partyId @return @throws FatalException	This method is used to get the details of all the parties depending on the account number It gives the details of parties including holder name,partyAccountRelationType,nickname etc. @param sessionContext @param account @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBySpidAndRoleId	This method fetches agent details from party module @param agentId @return @throws FatalException	This method invokes party service to fetch agent details using spid and agentType
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBySpidAndRoleId	This method fetches agent details from party module @param agentId @return @throws FatalException	@param sessionContext @param agentId @param agentType @return @throws FCRException @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllWarningIndicatorsAppliedOnParty	This method will fetch all the applied warning indicators for given party id	This method will fetch all the applied warning indicators for given party id
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllWarningIndicatorsAppliedOnParty	This method will fetch all the applied warning indicators for given party id	This method would fetch all the warning indicators for a given party and a corresponding event. @param eventCode @param partyId @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAllWarningIndicatorsAppliedOnParty	This method will fetch all the applied warning indicators for given party id	This method would fetch all the warning indicators for given party,event and the module type. @param partyId @param eventCode @param moduleType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	addApplicableWarningIndicator	This method will add all the applicable warning indicators for given warning indicator id	This method will add all the applicable warning indicators for given warning indicator id
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	addApplicableWarningIndicator	This method will add all the applicable warning indicators for given warning indicator id	This represents the addWarningIndicator method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,WarningIndicatorDTO,warningIndicatorDTO, ,warningIndicatorDTO TODO: Explain the warningIndicatorDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	addApplicableWarningIndicator	This method will add all the applicable warning indicators for given warning indicator id	This method adds the warning indicators defined for a party,  and if these warning indicators are to be applied,   it fetches the actions for the passed eventId. @param sessionContext @param eventId @param partyId @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	updateApplicableWarningIndicator	This method will update all the applicable warning indicators for given warning indicator id	This method will update all the applicable warning indicators for given warning indicator id

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	updateApplicableWarningIndicator	This method will update all the applicable warning indicators for given warning indicator id	This represents the updateWarningIndicator method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,WarningIndicatorDTO,warningIndicatorDTO, ,warningIndicatorDTO TODO: Explain the warningIndicatorDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	updateApplicableWarningIndicator	This method will update all the applicable warning indicators for given warning indicator id	This method adds the warning indicators defined for a party,  and if these warning indicators are to be applied,   it fetches the actions for the passed eventId. @param sessionContext @param eventId @param partyId @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	updateApplicableWarningIndicator	This method will update all the applicable warning indicators for given warning indicator id	This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchApplicableWarningIndicator	This method will fetch all the applicable warning indicators for given warning indicator id	This method has the following responsibility: <ol><li>1. Create query parameters from the export criteria.</li> <li>2. Calls fetchExportData to fetch DTO array.</li> <li>3. Create data handler from DTO.</li> <li>4. Create ExportReturn Details.</li></ol> @param sessionContext : {@link SessionContext} @param exportCriteria : ExportCriteria sent from FSM. @param exportControl : {@link ExportControlDetails} @return : {@link ExportReturnDetails} @throws FatalException : {@link FatalException} is raised only if Interaction.begin() and Interaction.close() fails.  If FatalException is raised, migration status and export data return is overridden in the calling method.
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchApplicableWarningIndicator	This method will fetch all the applicable warning indicators for given warning indicator id	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchApplicableWarningIndicator	This method will fetch all the applicable warning indicators for given warning indicator id	This represents the fetchWarningIndicator method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,WarningIndicatorDTO,warningIndicatorDTO, ,warningIndicatorDTO TODO: Explain the warningIndicatorDTO above @return {@link com.ofss.fc.app.account.dto.warningindicator.WarningIndicatorInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchApplicableWarningIndicator	This method will fetch all the applicable warning indicators for given warning indicator id	This method fetches the warning indicators defined for a party,  and if these warning indicators are to be applied,   it fetches the actions for the passed eventId. @param sessionContext @param eventId @param partyId @return @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	Sets the postal address details.Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	This method will check if the given party id is of party type individual,return true if party type is individual false otherwise @param sessionContext @param partyId @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	This method will check if the party is minor and the operating instruction channel access type is no for given channel then return true,false otherwise @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	This method will check if the party is operationally major and the operating instruction channel access type is no for given channel then return true,false otherwise @param partyAccountRelationshipDetailsDTO @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchPartyBasicDetails	This method will fetch the basic party details for the given party id @param sessionContext @param partyId @return	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRoleNameFromPartyToPartyRelationship	This method will fetch the account role name for given role code. @param roleCode @return	This method will fetch the account role name for given role code. @param roleCode @return
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRoleNameFromPartyToPartyRelationship	This method will fetch the account role name for given role code. @param roleCode @return	This method will set the account role name for the given account role,so if the party relation is SOW/JAF/JAO then the corresponding name of the enum is set in account role name else the name from account role of party is set in account role name. @param authorisedSignatoryPartyDTO
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRoleNameFromPartyToPartyRelationship	This method will fetch the account role name for given role code. @param roleCode @return	This method will set the account role name for the given account role,so if the party relation is SOW/JAF/JAO then the corresponding name of the enum is set in account role name else the name from account role of party is set in account role name. @param authorisedSignatoryPartyDTO
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	fetchAccountRoleNameFromPartyToPartyRelationship	This method will fetch the account role name for given role code. @param roleCode @return	This method will set the account role name for the given account role,so if the party relation is SOW/JAF/JAO then the corresponding name of the enum is set in account role name else the name from account role of party is set in account role name. @param authorisedSignatoryPartyDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	setPostalAddressDetails	Sets the postal address details. Gets the postal details as communication details for party. @param sessionContext the session context @param accountAddressLinkageDTO the account address linkage dto @throws FatalException the fatal exception	@param sessionContext @param loanAccountConfigurationDTO @param accountId @throws FatalException
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	checkIfWarningIndicatorsDefinedForPartiesOfAccount		This method is added to validate if warning indicators are applied for the party, and its associated parties.
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPartyAdapter	getCreateNewParty	This service will fetch the created partyId.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,PartyOnBoardingDTO, partyOnBoardingDTO, Set when a party (Individual/Corporate) is onboarded.  It holds all information about the new party. @return PartyOnBoardingResponse, Returns PartyId and its details. @throws com.ofss.fc.infra.exception.FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	getUpdatePartyData	This service will update the created party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,PartyGeneralInformationDTO,partyGeneralInformationDTO, Updates party(Individual/Corporate) data onboarded from AssociatedParty.  @return TransactionStatus, Returns update status of party. @throws com.ofss.fc.infra.exception.FatalException	
IAccountPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPartyAdapter	getUpdatePartyData	This service will update the created party. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,PartyGeneralInformationDTO,partyGeneralInformationDTO, Updates party(Individual/Corporate) data onboarded from AssociatedParty.  @return TransactionStatus, Returns update status of party. @throws com.ofss.fc.infra.exception.FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	notifyPoicyLinkage	The method will inform insurance provider of policy linkage with loan account. Once insurance provider receives the notification will activate the policy linkage at its end. @param facilityId @throws FatalException	This method returns the instance of LoanFTPAdapter. If isMockEnabled = True, LoanFTPAdapter with Mocked data is returned as a Response. If isMockEnabled = False, an instance of LoanFTPAdapter is returned as a response. Mocked Data is a dummy data. It is used to demonstrate the response from the Service from another module via this adapter.
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	notifyPoicyLinkage	The method will inform insurance provider of policy linkage with loan account. Once insurance provider receives the notification will activate the policy linkage at its end. @param facilityId @throws FatalException	This method is responsible to notify insurance of the policy linkage with loan account and on receiving this notification insurance provider will activate the policy linkage at its end. @param facilityId
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	isAnyActivePolicyLinked	The method will inquire insurance provider if there is any active policy attached to the given account. @param accountId @return	This method will find out if the given account can be closed.  The method will return true if   1. Account is not linked to any active policy.  2. The periodicity of the linked active policy is One-Time. @param accountId @return @throws FatalException
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	isAnyActivePolicyLinked	The method will inquire insurance provider if there is any active policy attached to the given account. @param accountId @return	The method is responsible to return amount to be refunded by insurance provider in case account is settled earlier than the maturity date. The method will  1. Check if given account is linked to any active policy.  2. If given account is linked to active policy, what is its premium paying period.  3. If periodicity of the policy is One-Time, inquire refund amount from insurance provider. @param accountId @param currencyCode @return
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchInsuranceDetails	The method is responsible to fetch details of the insurance policy for the given insurance id. @param insuranceId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchRefundedPremiumAmount	The method will fetch the amount be refunded by insurance provider if the loan account which linked to policy is being closed. @param accountId @return @throws FatalException	The method is responsible to return amount to be refunded by insurance provider in case account is settled earlier than the maturity date. The method will 1. Check if given account is linked to any active policy. 2. If given account is linked to active policy, what is its premium paying period. 3. If periodicity of the policy is One-Time, inquire refund amount from insurance provider. @param accountId @param currencyCode @return
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchRefundedPremiumAmount	The method will fetch the amount be refunded by insurance provider if the loan account which linked to policy is being closed. @param accountId @return @throws FatalException	<pre> For accounts where auto closure is Yes, in case of a direct payment in the account; a)Check if it is the last term loan account under that facility, if Yes; i. Consider the premium refund amount while arriving at the balance for auto closure ii. Compare the RPA balance with the amount required for auto closure, if it is sufficient then close the account. If it is insufficient, treat the payment as advance payment. Additional funds (after appropriating existing arrears) will remain in RPA iii. Refund amount will be displayed as a credit entry in the account with narration as 'premium amount refunded' before the final settlement entry b)If it not the last account under the facility then; Follow the auto closure process without considering the premium refund amount </pre> @param applicationContext @param accountId
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchRefundedPremiumAmount	The method will fetch the amount be refunded by insurance provider if the loan account which linked to policy is being closed. @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchExistingInsurancePremiumDetails	This method will fetch insurance details of the Proposed/Active policy linkage with given account based on value of linkageStatus. @param accountId @param linkageStatus @return @throws FatalException	<li><strong>inquirePolicyDetails()\A0-</strong> When account is getting disbursed we will call this method and use premium amount to derive amount for settlement and pass accounting accordingly.</li> @param accountId @param valueDate
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchExistingInsurancePremiumDetails	This method will fetch insurance details of the Proposed/Active policy linkage with given account based on value of linkageStatus. @param accountId @param linkageStatus @return @throws FatalException	The method will return insurance details attached to the account for given linkage status (Active or Proposed). @param accountId @param linkageStatus @return @throws FatalException
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchExistingInsurancePremiumDetails	This method will fetch insurance details of the Proposed/Active policy linkage with given account based on value of linkageStatus. @param accountId @param linkageStatus @return @throws FatalException	The method will return insurance details attached to the account for given linkage status (Active or Proposed). @param accountId @param linkageStatus @return @throws FatalException
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	fetchExistingInsurancePremiumDetails	This method will fetch insurance details of the Proposed/Active policy linkage with given account based on value of linkageStatus. @param accountId @param linkageStatus @return @throws FatalException	
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	isAnyProposedPolicyLinked		The method is responsible to check if a insurance policy is linked to the account but not yet activated. @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanInsuranceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanInsuranceAdapter	inquirePremiumQuote		The method will fetch premium quotes from insurance module. Will evenly distribute the premium amount per repayment stage and per repayment date. @param applicationContext @param accountScheduleAttributesDTO @param allRepaymentStages @param startingStageIndex @param onDate @return @throws FatalException
IIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method populates the Application level Attributes for CCI Welcome Letter Extract. It performs following operations : &lt;P&gt; 1.Takes sesionContext,OutboundDocumentRequestDTO as input &lt;P&gt; 2.Call populateApplicant to populate applicant level attributes &lt;P&gt; 3.Call populateProductGroup to populate populateProductGroup level attributes &lt;P&gt; 4.Appends all the retrieved elements to application level @param sesionContext @param outboundDocumentRequestDTO @return Application @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method Populating ProductGroups for CCI Welcome Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method is used to populate MainLine for product Group of CCI Welcome Letter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the CoverLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to CoverLetter to make the final CoverLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method populates the Application level Attributes for Cover Letter Extract. It performs following operations : &lt;P&gt; 1.Takes sesionContext,OutboundDocumentRequestDTO as input &lt;P&gt; 2.Call populateApplicant to populate applicant level attributes &lt;P&gt; 3.Call populateProductGroup to populate populateProductGroup level attributes &lt;P&gt; 4.Appends all the retrieved elements to application level @param sesionContext @param outboundDocumentRequestDTO @return Application @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method Populating ProductGroups for Cover Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method is used to populate MainLine for product Group of Cover Letter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the HypothecationLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to Hypothecation Letter to make the final HypothecationLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method populates the Application level Attributes for Hypothecation Letter Extract. It performs following operations : <P> 1.Takes sesionContext,OutboundDocumentRequestDTO as input <P> 2.Call populateApplicant to populate applicant level attributes <P> 3.Call populateProductGroup to populate populateProductGroup level attributes <P> 4.Appends all the retrieved elements to application level @param sesionContext @param lendingApplication @param outboundDocumentRequestDTO @return Application @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method Populating ProductGroups for HypothecationLetter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method is used to populate MainLine for product Group of HypothecationLetter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the PersonalGuaranteeLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to PersonalGuaranteeLetter to make the final PersonalGuaranteeLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method populates the Application level Attributes for PersonalGuarantee Letter Extract. It performs following operations : &lt;P&gt; 1.Takes sesionContext,OutboundDocumentRequestDTO as input &lt;P&gt; 2.Call populateApplicant to populate applicant level attributes &lt;P&gt; 3.Call populateProductGroup to populate populateProductGroup level attributes &lt;P&gt; 4.Appends all the retrieved elements to application level @param sesionContext @param lendingApplication @param outboundDocumentRequestDTO @return Application @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method Populating ProductGroups for PersonalGuaranteeLetter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method is used to populate MainLine for product Group of PersonalGuaranteeLetter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the MortgageLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to Mortgage Letter to make the final MortgageLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method populates the Application level Attributes for Mortgage Letter Extract. It performs following operations : <P> 1.Takes sesionContext,OutboundDocumentRequestDTO as input <P> 2.Call populateApplicant to populate applicant level attributes <P> 3.Call populateProductGroup to populate populateProductGroup level attributes <P> 4.Appends all the retrieved elements to application level @param sesionContext @param lendingApplication @param outboundDocumentRequestDTO @return Application @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method Populating ProductGroups for Mortgage Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method is used to populate MainLine for product Group of MortgageLetter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence &lt;P&gt; 1. Takes Domain Object as an input along with SessionContext. &lt;P&gt; 2. Populates the OfferLetter Node as the starting tag for the extract. &lt;P&gt; 3. Calls different modules to populate the requested attributes for the extract. &lt;P&gt; 4. Appends all the attribute to OfferLetter to make the final OfferLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method populates the Application level Attributes for Offer Letter Extract. It performs following operations : <P> 1.Takes sesionContext,OutboundDocumentRequestDTO as input <P> 2.Call populateApplicant to populate applicant level attributes <P> 3.Call populateProductGroup to populate populateProductGroup level attributes <P> 4.Appends all the retrieved elements to application level @param sesionContext @param outboundDocumentRequestDTO @return Application @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedApplication	<p>This service will inquire the application details for the application identifier passed. This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked, current application status, application type and the applicants that are attached to the application. In addition, simulation details attached to product groups for a given application are also inquired in this service. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application.</p>	<p>This method is used to populate Sub Line Account for Sub Line of Product Groups. @param context @param facilityId @param currency @param outboundDocumentRequestDTO @param creditFacilityDTO @return Account @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireOverDraftSimulationForSimulationId	<p>This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation.</p> <p><b>@fcb.param</b> MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. <b>@fcb.param</b> MI,String simulationId, , Indicates the identifier of the Simulation for which the selected simulation details are being fetched <b>@return</b> {@link ODSimulationResponse} This class contains the information that is retrieved for the simulation from draft for the term deposit account that is to be passed on during account opening. <b>@throws</b> FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationOriginationAdapter	inquireOverDraftSimulationForSimulationId	<p>This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String simulationId, , Indicates the identifier of the Simulation for which the selected simulation details are being fetched @return {@link ODSimulationResponse} This class contains the information that is retrieved for the simulation from draft for the term deposit account that is to be passed on during account opening. @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireOverDraftSimulationForSimulationId	<p>This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String simulationId, , Indicates the identifier of the Simulation for which the selected simulation details are being fetched @return {@link ODSimulationResponse} This class contains the information that is retrieved for the simulation from draft for the term deposit account that is to be passed on during account opening. @throws FatalException</p>	<p>This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireOverDraftSimulationForSimulationId	This service will inquire the Over Draft simulation details for the given simulationId. The simulation contains the account configuration details as filled out as per the applicant(s) preferences for the account as well as for each of the applicants in case of joint applications. These simulation details are available in the draft mode and are passed on to the Term Deposits module on account creation. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String simulationId, , Indicates the identifier of the Simulation for which the selected simulation details are being fetched @return {@link ODSimulationResponse} This class contains the information that is retrieved for the simulation from draft for the term deposit account that is to be passed on during account opening. @throws FatalException	
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	inquireDetailedSubmission	This service will inquire the submission that has already been submitted by the user. The submission will contain the details for all the applications within the submission and their corresponding	This service will inquire the submission details for the provided submission id.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>product group information. It also contains the applicant information for the applicants that are linked to the application. The simulation and account configuration details are also returned as a part of this service. @fcb.param  MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param  MI,String,submissionId, ,The id of the submission to be inquired  @return An instance of {@link SubmissionInquiryResponse} containing detailed information about the submission. @throws FatalException @exception &lt;br&gt;{@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} :  FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_HISTORY_NOT_FOUND}&lt;br&gt; &gt; FC_OR_DE_0075 error occurs when:&lt;br&gt; The submission history for the submission cannot be fetched.&lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIDetailsForApplication	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation. @param sessionContext @param applicationId @return @throws FatalException	
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIDetailsForApplication	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation. @param sessionContext @param applicationId @return @throws FatalException	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation.
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIDetailsForApplication	This service will inquire the cci details details for the given application. These details are provided for a lending application on account creation. @param sessionContext @param applicationId @return @throws FatalException	This method generates the data extract using the Request DTO parameters. It performs the following steps in sequence <P> 1. Takes Domain Object as an input along with SessionContext. <P> 2. Populates the OfferLetter Node as the starting tag for the extract. <P> 3. Calls different modules to populate the requested attributes for the extract. <P> 4. Appends all the attribute to OfferLetter to make the final OfferLetter Extract. @param SessionContext :Contains basic Context level attributes. @param DomainObjectDTO :Contains basic key attribute to call different modules. @return IExtract. @throws FatalException.
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionContext @param insuranceId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionId @param insuranceId @return @throws FatalException	This service will fetch CCI premium quote amount for an Insurance on basis of InsuranceID.
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionId @param insuranceId @return @throws FatalException	This method will fetch ApplicantProductGroupReferenceDTOs for given insuranceId.
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionId @param insuranceId @return @throws FatalException	This method populates the Application level Attributes for Offer Letter Extract. It performs following operations : <P> 1.Takes sessionId,OutboundDocumentRequestDTO as input <P> 2.Call populateApplicant to populate applicant level attributes <P> 3.Call populateProductGroup to populate populateProductGroup level attributes <P> 4.Appends all the retrieved elements to application level @param sessionId @param outboundDocumentRequestDTO @return Application @throws FatalException
IntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationOriginationAdapter	fetchCCIPolicyDetails	This service will fetch CCI Premium amount details for a given insurance id. @param sessionId @param insuranceId @return @throws FatalException	This method Populating ProductGroups for Offer Letter. @param sessionId @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	inquireSubmissionDecision	@param sessionContext @param submissionId @return @throws FatalException	
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	inquireSubmissionDecision	@param sessionContext @param submissionId @return @throws FatalException	This method is used to populate MainLine for product Group of Offer Letter. @param sessionContext @param loanProductMainLineLinkageDTO @param outboundDocumentRequestDTO @return MainLine @throws FatalException
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchSubscriptionsDetails	@param sessionContext @param bundleReferenceId This service will fetch bundle details for given bundle reference id. This contains all the subscription details for application.	
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchSubscriptionsDetails	@param sessionContext @param bundleReferenceId This service will fetch bundle details for given bundle reference id. This contains all the subscription details for application.	
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchCCIPolicyDetailsForInsuranceId	@param insuranceId @return ApplicantProductGroupReferenceDTO[] @throws FatalException This method will fetch list of applicant product group reference for given insuranceId.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchCCIPolicyDetailsForInsuranceId	@param insuranceId @return ApplicantProductGroupReferenceDTO[] @throws FatalException This method will fetch list of applicant product group reference for given insuranceId.	This method will fetch ApplicantProductGroupReferenceDTOs for given insuranceId.
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchCCIPolicyDetailsForInsuranceId	@param insuranceId @return ApplicantProductGroupReferenceDTO[] @throws FatalException This method will fetch list of applicant product group reference for given insuranceId.	This method populates the Application level Attributes for Offer Letter Extract. It performs following operations : <P> 1.Takes sessionContext,OutboundDocumentRequestDTO as input <P> 2.Call populateApplicant to populate applicant level attributes <P> 3.Call populateProductGroup to populate populateProductGroup level attributes <P> 4.Appends all the retrieved elements to application level @param sessionContext @param outboundDocumentRequestDTO @return Application @throws FatalException
IIIntegrationOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIIntegrationOriginationAdapter	fetchCCIPolicyDetailsForInsuranceId	@param insuranceId @return ApplicantProductGroupReferenceDTO[] @throws FatalException This method will fetch list of applicant product group reference for given insuranceId.	This method Populating ProductGroups for Offer Letter. @param sessionContext @param loanProductLinkage @param outboundDocumentRequestDTO @return ProductGroup @throws FatalException
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	getAccrualStatusHistory	This method returns a map with the date range as key and value as AccrualStatus for the date range. This would return this map for all accrual status changes that occurred for the account. @param accountId @return Map<Range<Date>, AccrualStatusType> @throws FatalException	This method returns a map with the date range as key and value as AccrualStatus for the date range. This would return this map for all accrual status changes that occurred for the account. @param accountId @return Map<Range<Date>, AccrualStatusType>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	getAccrualStatusHistory	This method returns a map with the date range as key and value as AccrualStatus for the date range. This would return this map for all accrual status changes that occurred for the account. @param accountId @return Map<Range<Date>, AccrualStatusType> @throws FatalException	This method gets the accrual status on the date passed for the account. @param accountId @return Map<Range<Date>, AccrualStatusType>
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForNewAccount	This method will be called when ever a new loan account is created. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param appContext @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @param branchId @return accountClassificationResponse	This method will be called when ever a new loan account is created. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @return accountClassificationResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForNewAccount	This method will be called when ever a new loan account is created. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param appContext @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @param branchId @return accountClassificationResponse	
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForNewAccount	This method will be called when ever a new loan account is created. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param appContext @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @param branchId @return accountClassificationResponse	<pre>This method calls the NPA module to notify For New Account. NPA module returns the accrual status and classification code for newly opened account.</pre> @param applicationContext @param accountId @param loanAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForAccountUpdate	This method will be called when ever the facilityID or Party ID of an account is changed. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param appContext @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @return accountClassificationResponse	This method will be called when ever the facilityID or Party ID of an account is changed. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @return accountClassificationResponse @throws FatalException
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForAccountUpdate	This method will be called when ever the facilityID or Party ID of an account is changed. The relevant details about the new account are sent to NPA module. NPA module returns the CRR code and accrual status in AccountClassificationResponse.actualCodeDetails @param appContext @param accountId @param productCode @param currencyCode @param businessUnit @param partyId @param facilityId @return accountClassificationResponse	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId @param accountStatus	Gets current provision amount from asset classification. Passes accounting for provision reversal. Updates provision drawn amount(same as current provision amount) to asset classification. Updates account status to asset classification as closed. @param account
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId @param accountStatus	This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId @param accountStatus
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId @param accountStatus	
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId	Gets current provision amount from asset classification. Passes accounting for provision reversal. Updates provision drawn amount(same as current provision amount) to asset classification. Updates account status to asset classification as closed. @param account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param accountStatus	
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAssetClassificationPreference		This method is to be used to notify AssetClassification module whenever the status of account changes to Closed. Account Module not Loan Module will notify NPA if account is fully written off. Partial write-off is not considered for notification. @param accountId @param accountStatus
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAssetClassificationPreference		
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAssetClassificationPreference		
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForDPDFreez	@param accountId @param reliefStartDate @param reliefExpiryDate	@param accountId @param reliefStartDate @param reliefExpiryDate
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForDPDFreez	@param accountId @param reliefStartDate @param reliefExpiryDate	This method is used to notify DPD freez to Assest classsication module. @param loanHardshipDetails

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param transactionType @param arrearType @param arrearAmount	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param transactionType @param arrearType @param arrearAmount	This method will call Asset classification to notify it when ever any arrear is raised or paid. @param accountId @param transactionType @param arrearType @param arrearAmount
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param transactionType @param arrearType @param arrearAmount	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountAssetClassification	<p>&lt;p&gt;This method is responsible for fetching Asset Classification Status of a Loan Account.&lt;br&gt;This method is specifically used to log the details required for Dashboard Notification while updating CRR Code.&lt;br&gt; The basic purpose is to identify the following scenarios:-&lt;br&gt; &lt;ul&gt; &lt;li&gt;CRR Code update: NORMAL -&gt; NORMAL&lt;/li&gt; &lt;li&gt;CRR Code Update: SUSPENDED -&gt; SUSPENDED&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;&lt;br&gt; When CRR Code is updated (assuming the Asset Classification Status is unchanged as SUSPENDED), Dashboard notification need to be sent.&lt;br&gt; This scenario is referred to treat a Loan Account as a Non Performing Asset.&lt;br&gt; &lt;br&gt; &lt;b&gt;&lt;u&gt;NOTE:-&lt;/u&gt;&lt;/b&gt;&lt;br&gt;This method is exclusively referred by {&lt;link LoanArrearManagementApplication#updateAccountCRRCode(ApplicationContext, String, String)} to know the present Asset Classification Status of Loan Account.&lt;br&gt; &lt;/p&gt; @param accountId @return</p>	<p>This method returns a map with the date range as key and value as AccrualStatus for the date range. This would return this map for all accrual status changes that occurred for the account. @param accountId @return Map&lt;Range&lt;Date&gt;, AccrualStatusType&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountAssetClassification	<p>&lt;p&gt;This method is responsible for fetching Asset Classification Status of a Loan Account.&lt;br&gt;This method is specifically used to log the details required for Dashboard Notification while updating CRR Code.&lt;br&gt; The basic purpose is to identify the following scenarios:-&lt;br&gt; &lt;ul&gt; &lt;li&gt;CRR Code update: NORMAL -&gt; NORMAL&lt;/li&gt; &lt;li&gt;CRR Code Update: SUSPENDED -&gt; SUSPENDED&lt;/li&gt; &lt;/ul&gt;&lt;/li&gt;&lt;br&gt; When CRR Code is updated (assuming the Asset Classification Status is unchanged as SUSPENDED), Dashboard notification need to be sent.&lt;br&gt; This scenario is referred to treat a Loan Account as a Non Performing Asset.&lt;br&gt; &lt;br&gt; &lt;b&gt;&lt;u&gt;NOTE:-&lt;/u&gt;&lt;/b&gt;&lt;br&gt;This method is exclusively referred by {@link LoanArrearManagementApplication#updateAccountCRRCode(ApplicationContext, String, String)} to know the present Asset Classification Status of Loan Account.&lt;br&gt; &lt;/p&gt; @param accountId @return</p>	<p>&lt;p&gt;This method is responsible for fetching Asset Classification Status of a Loan Account.&lt;br&gt;This method is specifically used to log the details required for Dashboard Notification while updating CRR Code.&lt;br&gt; The basic purpose is to identify the following scenarios:-&lt;br&gt; &lt;ul&gt; &lt;li&gt;CRR Code update: NORMAL -&gt; NORMAL&lt;/li&gt; &lt;li&gt;CRR Code Update: SUSPENDED -&gt; SUSPENDED&lt;/li&gt; &lt;/ul&gt;&lt;/li&gt;&lt;br&gt; When CRR Code is updated (assuming the Asset Classification Status is unchanged as SUSPENDED), Dashboard notification need to be sent.&lt;br&gt; This scenario is referred to treat a Loan Account as a Non Performing Asset.&lt;br&gt; &lt;br&gt; &lt;b&gt;&lt;u&gt;NOTE:-&lt;/u&gt;&lt;/b&gt;&lt;br&gt;This method is exclusively referred by {@link LoanArrearManagementApplication#updateAccountCRRCode(ApplicationContext, String, String)} to know the present Asset Classification Status of Loan Account.&lt;br&gt; &lt;/p&gt; @param accountId @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountAssetClassification	<p>&lt;p&gt;This method is responsible for fetching Asset Classification Status of a Loan Account.&lt;br&gt;This method is specifically used to log the details required for Dashboard Notification while updating CRR Code.&lt;br&gt; The basic purpose is to identify the following scenarios:-&lt;br&gt; &lt;ul&gt; &lt;li&gt;CRR Code update: NORMAL -&gt; NORMAL&lt;/li&gt; &lt;li&gt;CRR Code Update: SUSPENDED -&gt; SUSPENDED&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;&lt;br&gt; When CRR Code is updated (assuming the Asset Classification Status is unchanged as SUSPENDED), Dashboard notification need to be sent.&lt;br&gt; This scenario is referred to treat a Loan Account as a Non Performing Asset.&lt;br&gt; &lt;br&gt; &lt;b&gt;&lt;u&gt;NOTE:-&lt;/u&gt;&lt;/b&gt;&lt;br&gt;This method is exclusively referred by {&lt;a href="#"&gt;link LoanArrearManagementApplication#updateAccountCRRCode(ApplicationContext, String, String)} to know the present Asset Classification Status of Loan Account.&lt;br&gt; &lt;/p&gt; <p>@param accountId @return</p> </p>	<p>This method will be called from LoanAccountApplication to fetch DPD for an account. @param accountId @return dpd</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountAssetClassification	<p>This method is responsible for fetching Asset Classification Status of a Loan Account. This method is specifically used to log the details required for Dashboard Notification while updating CRR Code. The basic purpose is to identify the following scenarios:-</p> <ul style="list-style-type: none"> <li>CRR Code update: NORMAL -&gt; NORMAL</li> <li>CRR Code Update: SUSPENDED -&gt; SUSPENDED</li> </ul> <p>When CRR Code is updated (assuming the Asset Classification Status is unchanged as SUSPENDED), Dashboard notification need to be sent. This scenario is referred to treat a Loan Account as a Non Performing Asset.</p> <p><b>NOTE:-</b> This method is exclusively referred by <a href="#">LoanArrearManagementApplication#updateAccountCRRCode(ApplicationContext, String, String)</a> to know the present Asset Classification Status of Loan Account.</p> <p>@param accountId @return</p>	<p>This method is called to set the account's new CRR Code. This method should be called when during classification batch the CRR code for an account is being changed but the accrual status is unchanged.</p> <p><b>NOTE:-</b> Notification Details will be logged Only when ACCRUAL Status is SUSPENDED and CRR Code is being updated.</p> <p>@param applicationContext @param accountId @param crrCode</p>
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchDPDForAccount	<p>This method will be called from LoanAccountApplication to fetch DPD for an account.</p> <p>@param accountId @return dpd</p>	<p>This method will be called from LoanAccountApplication to fetch DPD for an account.</p> <p>@param accountId @return dpd</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchDPDForAccount	This method will be called from LoanAccountApplication to fetch DPD for an account. @param accountId @return dpd	This method is responsible for fetching the basic details of the   Loan Account . The inputs to the method is accountId .   @param accountId @return LoanAccountCommonInformationDTO
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountWriteOffHistory	@param accountId @return @throws FatalException	@param accountId @param backValueDate @param eventCode @param maintenanceType @return
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountWriteOffHistory	@param accountId @return @throws FatalException	This method checks whether the RateChange action is eligible to be logged in the Actions table for processing.    It checks for two eligibility conditions:     Eligibility Condition 1: Check if any rollover instruction exists for the RepriceDate. If a rollover instruction exists for the repriceDate, then the RateChange will be handled by LoanRolloverService. The Pricing related RateChange action would be inEligible and so no RateChange action needs to be logged in the Actions table.    Eligibility Condition 2: Check if the stage in which the Reprice is occurring is in a fixed-rate period. If the stage in which the Reprice is occurring is in a fixed-rate period, then the Pricing related RateChange action would be inEligible and so no RateChange action needs to be logged in the Actions table. @param accountId @param priceType @param repriceDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccountWriteOffHistory	@param accountId @return @throws FatalException	This method will validate and throws business exception if following conditions are not fulfilled. <pre> &lt;strong&gt;Display error if the UDM start date is: &lt;/strong&gt; &lt;li&gt;Before First Disbursement Date. &lt;/li&gt; &lt;li&gt;Before Partial Write Off date &lt;/li&gt; &lt;li&gt;Before Full Write Off date. &lt;/li&gt; &lt;li&gt;Before Loan Recession. &lt;/li&gt; &lt;li&gt;Before Conversion date (i.e. Movement from Legacy to New System) &lt;/li&gt; &lt;li&gt;Back value date should not be a holiday. &lt;/li&gt; &lt;li&gt;Account should not be in fixed rate on back value date. &lt;/li&gt; &lt;/pre&gt; @param accountId @param varainceStartDate</pre>
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	recordRecoveryForWriteOffClosed	This method is responsible for recording the payments received towards a Loan account which is either in CLOSED or WRITTEN_OFF status. If there are any unrecovered arrears on the Account, this method will appropriate against the arrears. If excess amount is received, it will be appropriated against LN_EXCESS_PAYMENT_RECEIVED_SUSPENSE GL. @param applicationContext @param accountId @param transactionDetailsDTO @param eventSequenceNumber TODO @param productCode TODO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	recordRecoveryForWrittenOffClosed	<p>This method is responsible for recording the payments received towards a Loan account which is either in CLOSED or WRITTEN_OFF status. If there are any unrecovered arrears on the Account, this method will appropriate against the arrears. If excess amount is received, it will be appropriated against LN_EXCESS_PAYMENT_RECEIVED_SUSPENSE GL. @param applicationContext @param accountId @param transactionDetailsDTO @param eventSequenceNumber TODO @param productCode TODO @return</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccrualStatusForEffectiveDate	<p>This method is to be used to fetch the accrual status of an account for the given date. &lt;ul&gt; &lt;li&gt;Returns new classification code in cases where there is either a batch or manual classification. &lt;br&gt; Example: If on 9-Apr-2013, the account moves from [101-Normal] to [103-Suspended] status, [103-Suspended] is returned as the classification code for 9-Apr-2013.&lt;/li&gt; &lt;li&gt;Returns default code &lt;b&gt;'AC000'&lt;/b&gt;: If the given date is lesser than first classification movement to a different Classification code. This could happen for newly opened accounts or for dates earlier than account opening date.&lt;/li&gt; &lt;li&gt;Returns current classification code if the given date is earlier than current date and is greater than the last classification movement to a different code.&lt;/li&gt;</p>	<p>This method is to be used to fetch the accrual status of an account for the given date. &lt;ul&gt; &lt;li&gt;Returns new classification code in cases where there is either a batch or manual classification. &lt;br&gt; Example: If on 9-Apr-2013, the account moves from [101-Normal] to [103-Suspended] status, [103-Suspended] is returned as the classification code for 9-Apr-2013.&lt;/li&gt; &lt;li&gt;Returns default code &lt;b&gt;'AC000'&lt;/b&gt;: If the given date is lesser than first classification movement to a different Classification code. This could happen for newly opened accounts or for dates earlier than account opening date.&lt;/li&gt; &lt;li&gt;Returns current classification code if the given date is earlier than current date and is greater than the last classification movement to a different code.&lt;/li&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanAssetClassificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanAssetClassificationAdapter	fetchAccrualStatusForEffectiveDate	This method is to be used to fetch the accrual status of an account for the given date. <ul> <li>Returns new classification code in cases where there is either a batch or manual classification.   Example: If on 9-Apr-2013, the account moves from [101-Normal] to [103-Suspended] status, [103-Suspended] is returned as the classification code for 9-Apr-2013.</li> <li>Returns default code <b>'AC000'</b>: If the given date is lesser than first classification movement to a different Classification code. This could happen for newly opened accounts or for dates earlier than account opening date.</li> <li>Returns current classification code if the given date is earlier than current date and is greater than the last classification movement to a different code.</li>	@param accountId @param backValueDate @param eventCode @param maintenanceType @return
IInsuranceRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceRulesAdapter	executeRule	This method executes a rule and gives the outcome of the rule. @fcb.param sessionContext @fcb.param ruleId @fcb.param factDTOs @return @throws FatalException	This service will execute the rule for given rule id and facts. @param sessionContext , ruleId, factDTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceRulesAdapter	executeRule	This method executes a rule and gives the outcome of th rule. @fcb.param sessionId @fcb.param ruleId @fcb.param factDTOs @return @throws FatalException	
IInsuranceRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsuranceRulesAdapter	executeRule	This method executes a rule and gives the outcome of th rule. @fcb.param sessionId @fcb.param ruleId @fcb.param factDTOs @return @throws FatalException	
IAccountingPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingPartyAdapter	getDeterminantValue		This method is used to return the determinant value of party @param partyId @return
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyRelationDetailForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionId This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} objects containing party and account relationship details @throws FatalException	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionId This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} Objects containing party and account relationship details @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyRelationDetailForAccount(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyRelationDetailForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} objects containing party and account relationship details @throws FatalException	This method returns the joint account holder details @param accountId @return PartyAccountRelationshipDetails[] @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyRelationDetailForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} objects containing party and account relationship details @throws FatalException	This method creates a New Account in the Product specified by PMU. It loads the existing Account and sets the Product Code to the Reinvest Product and carries forward the existing Account Attributes to the New Account. This method is called when Product level parameter is set to Reinvest at Maturity and Deposit level instructions have not been maintained. @param applicationContext @param reinvestProductCode @param deposit @param iTermDepositAccount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyRelationDetailForAccount	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} objects containing party and account relationship details @throws FatalException	This method creates a New Account in the Product specified by PMU. It loads the existing Account and sets the Product Code to the Reinvest Product and carries forward the existing Account Attributes to the New Account. This method is called when Product level parameter is set to Reinvest at Maturity and Deposit level instructions have not been maintained. @param applicationContext @param reinvestProductCode @param deposit @param iTermDepositAccount @return
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	createPartyAccountRelationship	This method inserts a party and account relationship. Basically, any customer can have multiple accounts, or multiple accounts can have same party Id. @param partyAccountRelationshipDetails Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details	This method inserts a party and account relationship. Basically, any customer can have multiple accounts, or multiple accounts can have same party Id. @param partyAccountRelationshipDetails Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#createPartyAccountRelationship(PartyToAccountRelationshipDTO[])
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	createPartyAccountRelationship	This method inserts a party and account relationship. Basically, any customer can have multiple accounts, or multiple accounts can have same party Id. @param partyAccountRelationshipDetails Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details	This method is used to open an account. Algo 1. Get the party KYC status. If it is confirmed then mark the account status as confirm else mark the account as AccountStatus.KYCCunconfirmed 2. Get the generated Account Number from account generation method 3. Call the create method of maintenanceDomainService to persist 4. Log account closure action 5. If account has been sourced by broker/introducer, then link agent details to an account

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	createPartyAccountRelationship	This method inserts a party and account relationship. Basically, any customer can have multiple accounts,   or multiple accounts can have same party Id. @param partyAccountRelationshipDetails Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details	This method is used to create party account relationship @param partyAccountRelationshipDetails
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	createPartyAccountRelationship	This method inserts a party and account relationship. Basically, any customer can have multiple accounts,   or multiple accounts can have same party Id. @param partyAccountRelationshipDetails Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details	This method will open an account from existing account and new product. @param sessionContext @param accountId @param productId @return TermDepositAccountResponse @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This method validate the account status based on the Primary party.   <table border=1> <th>Sr No</th> <th>Primary Party Minor-Major Status</th> <th>Secondary Party Available</th> <th>Secondary Party Minor-Major Status</th> <th>Account Opening Permitted</th> <th>TD Account Major-Minor Status</th></tr> <tr> <td>1</td> <td>Minor</td> <td>No</td> <td>NA</td> <td>Y</td> <td>Minor</td> </tr> <tr> <td>2</td> <td>Minor</td> <td>Yes</td> <td>Minor</td> <td>Y</td> <td>Minor</td> </tr> <tr> <td>3</td> <td>Minor</td> <td>Yes</td> <td>Operational Major</td> <td>Y</td> <td>Minor</td> </tr> <tr> <td>4</td> <td>Minor</td>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;5&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;No&lt;/td&gt; &lt;td&gt;NA&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;6&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;7&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;8&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;No&lt;/td&gt; &lt;td&gt;NA&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;9&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;10&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;11&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;/table&gt; </pre>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	<p>a) The status change from minor--&gt; Legal major is done based on whether the &lt;br&gt; flag set at bank level is manual or automatic. In case of Flag set at Product level is manual , &lt;br&gt; this field is allowed to be modified by the OE else it will be a display field. &lt;br&gt;</p> <p>b) The minor can become legal major depending on the bank level set-up. &lt;br&gt; However after becoming legal major the status cannot be changed to minor. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyDetails(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This method gives the age status i.e party is Minor/Major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getAgeStatus(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchPartiesOfTdAccount(com.ofss.fc.app.context.SessionContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyDetails(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			information about party	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This method returns if any of the party is black listed.  If the Blacklisting status of any of the holders is $\cap \gamma \Downarrow$ Blacklisted $\cap \gamma \Downarrow$ , Throw "TD account opening is not permitted Party is blacklisted".  @param partyId @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party	This method checks if the party is deceased.   It returns true if customer is $\cap \gamma \Downarrow$ Deceased $\cap \gamma \Downarrow$ else false.  @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or	This method validate the account status based on the Primary party.  <table border=1> <th>Sr No</th> <th>Primary Party Minor-Major Status</th> <th>Secondary Party Available</th> <th>Secondary Party Minor-Major Status</th> <th>Account Opening Permitted</th> <th>TD Account Major-Minor Status</th></tr> <tr> <td>1</td> <td>Minor</td> <td>No</td> <td>NA</td> <td>Y</td> <td>Minor</td> </tr>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			minor	<pre> &lt;br&gt; &lt;tr&gt; &lt;td&gt;2&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;3&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;4&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;5&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;No&lt;/td&gt; &lt;td&gt;NA&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;6&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;7&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;8&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;No&lt;/td&gt; &lt;td&gt;NA&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;9&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Minor&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;10&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Operational Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;tr&gt; &lt;td&gt;11&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Yes&lt;/td&gt; &lt;td&gt;Legal Major&lt;/td&gt; &lt;td&gt;Y&lt;/td&gt; &lt;td&gt;Major&lt;/td&gt; &lt;/tr&gt; &lt;br&gt; &lt;/table&gt; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	a) The status change from minor---> Legal major is done based on whether the   flag set at bank level is manual or automatic. In case of Flag set at Product level is manual ,   this field is allowed to be modified by the OE else it will be a display field.   b) The minor can become legal major depending on the bank level set-up.   However after becoming legal major the status cannot be changed to minor.  
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	This method assembles the partyDetails from CustomerDetailsInquiryResponse @param customerDetailsInquiryResponse {@link CustomerDetailsInquiryResponse} Object having customer details which will be used to get data for assembling party details object @param partyDetails {@link Partydetails} Object which needs to be assembled
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	This method gives the age status i.e party is Minor/Major  @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getAgeStatus(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	This method returns the AgeStatus given the partyId with role as Customer @param partyId @return @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatus	This method fetches status of the age of party which specifies whether the party is minor or major @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyEmployee	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag PartyEmployee need to be fetched @return boolean	If the employee flag is true and the party does not have employee role then throw exception @param isEmployeeAccount @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyEmployee	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag PartyEmployee need to be fetched @return boolean	This method gives the age status i.e party is Minor/Major  @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getAgeStatus(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyEmployee	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag PartyEmployee need to be fetched @return boolean	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag PartyEmployee need to be fetched @return boolean @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#isPartyEmployee(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyEmployee	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag PartyEmployee need to be fetched @return boolean	This method checks if the party is employee.   It returns true if customer is $\cap$ Employee $\cap$ else false.  @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isCustomerRoleAssigned	This method return true if the party has customer role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag CustomerRoleAssigned need to be fetched @return boolean	@param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isCustomerRoleAssigned	This method return true if the party has customer role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag CustomerRoleAssigned need to be fetched @return boolean	This method return true if the party has employee role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag CustomerRoleAssigned need to be fetched @return boolean com.ofss.fc.app.adapter.td.ITDPartyAdapter#isCustomerRoleAssigned(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isCustomerRoleAssigned	This method return true if the party has customer role else false @param applicationContext Parameter containing information about the system @param partyId The party for which the flag CustomerRoleAssigned need to be fetched @return boolean	This method checks if the party is employee.   It returns true if customer is $\cap \uparrow \downarrow$ Employee $\cap \uparrow \downarrow$ else false.  @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	This method constructs the TD account opening DTO @param sessionContext @param accDTO @param applicationId @return TermDepositAccountOpenDTO @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	This method constructs the TD account opening DTO @param sessionContext @param accountConfigDTO @param applicationId @return TermDepositAccountOpenDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	This method populates the common fields required for alerts processing of all events  <li>Deposit related information</li> @param accountId @param depositId @return DepositEventDetailDTO
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchPartyName(java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	If the party title is null, set it to the title from the previous arrangement @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyName	This method will return the party's full name for the party identifier provided @param partyId The party for which the full name need to be fetched @return String, ,Contains the party's full name @throws FatalException	This method fetches term deposit account details and party name. @param accountId term deposit account id @return {@link TermDepositAccountDetailsDTO} Object containing account details to be displayed on premature redeem td calculator screen.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartiesOfTdAccount	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException	This method fills the following details in the activityLog {@link TDFinancialEventDetailDTO} <ul><li>AccountId : The account that is being transferred</li> <li>AccountHolderName : The name of the account holder</li> <li>ExistingBranchName : The branch from where the account is transferred</li> <li>newBranchName : The branch name to which the account is transferred</li> </ul> @param accountId @return @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartiesOfTdAccount	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException	This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relation ship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartiesOfTdAccount	<p>This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException</p>	<p>This method will insert/update the account operating instructions for account when secondary applicant is added to the account from origination. @param sessionContext @param accountId @param holderPreferenceDTO @param operatingInstructionCategoryType @param individualPartyRole @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartiesOfTdAccount	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchPartiesOfTdAccount(com.ofss.fc.app.context.SessionContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartiesOfTdAccount	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchPartiesOfTdAccount(com.ofss.fc.app.context.SessionContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	inquirePartyDetails	This service will get the partyDetailsDTO @param partyId The party for which party details need to be fetched @return {@link PartyDetailsDTO} dto object containing party details	This service will get the partyDetailsDTO @param partyId The party for which party details need to be fetched @return {@link PartyDetailsDTO} DTO object containing party details @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#inquirePartyDetails(java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	inquirePartyDetails	This service will get the partyDetailsDTO @param partyId The party for which party details need to be fetched @return {@link PartyDetailsDTO} dto object containing party details	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyDetails(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	inquirePartyDetails	This service will get the partyDetailsDTO @param partyId The party for which party details need to be fetched @return {@link PartyDetailsDTO} dto object containing party details	The method fetches the following parameters Branch Name, Product Name ,Party Name @param termDepositAccountDTO @return
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	changePartyAccountRelationship	This service will change the party to account relationship type for each party provided as input @param sessionContext This parameter captures the basic information about the user of the system. @param partyToAccountRelationshipDTO Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details @throws FatalException	This method is used update the Account Holder preferences for a Term Deposit Account that has already been opened previously. In case of joint applications where one or more applicant(s) are not present, each applicant provides preferences such as statement and address for delivery. The new joint applicant is also added in the party account relationship.  If the account is opened with a single applicant, the single party is initially added as a Sole Owner on the account. When a joint applicant provides preferences for such account , the existing primary party relationship is changed to Joint and First and the new joint applicant is added as Joint and Other.  For accounts opened with 2 or more applicants, the existing party relationships remain unchanged as Joint And First and Joint And Other. The new joint applicant is added with a relationship of Joint And Other.  Finally, all joint applicant party level statement preferences and addresses are updated. @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String accountId, Indicates the identifier for the account for which account holder preferences are being updated. @fcb.param MI,TDAccountHolderPreferenceDTO,tdAccountHolderPreferenceDTO, , This class represents the DTO containing account holder preferences for term deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>accounts.&lt;br&gt; This class contains information about preferences captured for the applicant of the term deposit account. It has information about the external banking facilities such as internet banking, phone banking and mobile banking, nickname provided for each deposit only in case of multiple deposits within a single account and term deposit letter delivery preferences. @fcb.param MI, Boolean, isUpdateAccountRelationshipRequired, , Indicates if account relationship is to be updated. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException</p>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	changePartyAccountRelationship	<p>This service will change the party to account relationship type for each party provided as input @param sessionContext This parameter captures the basic information about the user of the system. @param partyToAccountRelationshipDTO Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details @throws FatalException</p>	<p>This service will change the party to account relationship type for each party provided as input @param sessionContext This parameter captures the basic information about the user of the system. @param partyToAccountRelationshipDTO Array of {@link PartyToAccountRelationshipDTO} objects containing party account relationship details @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#changePartyAccountRelationship(com.ofss.fc.app.context.SessionContext, PartyToAccountRelationshipDTO[])</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.domain.party.entity.relation.account.PartyAccountRelationship} @throws FatalException	This method returns specific customer details on the basis of the Account Id passed. It returns party name, party id and ownership status @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return Array of {@link PartyToAccountRelationshipDTO} Objects containing party and account relationship details @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyRelationDetailForAccount(java.lang.String, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.domain.party.entity.relation.account.PartyAccountRelationship} @throws FatalException	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchAllPartiesOfTdAccount(com.ofss.fc.app.context.SessionContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.domain.party.entity.relation.account.PartyAccountRelationship} @throws FatalException	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.app.party.dto.relation.account} @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchAllPartiesOfAccount(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.domain.party.entity.relation.account.PartyAccountRelationship} @throws FatalException	Returns all the parties related to the accountId Input toThis method is the account Id. @param sessionContext @param accountId @return @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllPartiesOfAccount	Returns all the parties related to the Account @param accountId Term deposit account id @return Array of {@link com.ofss.fc.domain.party.entity.relation.account.PartyAccountRelationship} @throws FatalException	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchActivePartyMemos	The method checks if memo is present on a given party @param partyId Identifies a partyId for which the memo is to be checked. @return {@link CommentInquiryResponse} Signifies the memo information present on the party of the TD account	The method checks if the memo is maintained for the party of the TD account and if present,  the level of severity is logged for overridable business policy check.
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method validate the account status based on the Primary party.  <table border=1> <tr><th>Sr No</th> <th>Primary Party Minor-Major Status</th> <th>Secondary Party Available</th> <th>Secondary Party Minor-Major Status</th> <th>Account Opening Permitted</th> <th>TD Account Major-Minor Status</th></tr> <tr><td>1</td> <td>Minor</td> <td>No</td> <td>NA</td> <td>Y</td> <td>Minor</td> </tr> <tr><td>2</td> <td>Minor</td> <td>Yes</td> <td>Minor</td> <td>Y</td> <td>Minor</td> </tr> <tr><td>3</td> <td>Minor</td> <td>Yes</td> <td>Operational

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Major</td> <td>Y</td> <td>Minor</td> </tr>   <tr> <td>4</td> <td>Minor</td> <td>Yes</td> <td>Legal Major</td> <td>Y</td> <td>Minor</td> </tr>   <tr> <td>5</td> <td>Operational Major</td> <td>No</td> <td>NA</td> <td>Y</td> <td>Operational Major</td> </tr>   <tr> <td>6</td> <td>Operational Major</td> <td>Yes</td> <td>Operational Major</td> <td>Y</td> <td>Operational Major</td> </tr> <tr> <td>7</td> <td>Operational Major</td> <td>Yes</td> <td>Legal Major</td> <td>Y</td> <td>Operational Major</td> </tr>   <tr> <td>8</td> <td>Legal Major</td> <td>No</td> <td>NA</td> <td>Y</td> <td>Major</td> </tr>   <tr> <td>9</td> <td>Legal Major</td> <td>Yes</td> <td>Minor</td> <td>Y</td> <td>Major</td> </tr>   <tr> <td>10</td> <td>Legal Major</td> <td>Yes</td> <td>Operational Major</td> <td>Y</td> <td>Major</td> </tr>   <tr> <td>11</td> <td>Legal Major</td> <td>Yes</td> <td>Legal Major</td> <td>Y</td> <td>Major</td> </tr>   </table>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	a) The status change from minor--> Legal major is done based on whether the   flag set at bank level is manual or automatic. In case of Flag set at Product level is manual ,   this field is allowed to be modified by the OE else it will be a display field.   b) The minor can become legal major depending on the bank level set-up.   However after becoming legal major the status cannot be changed to minor.    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method fetches party details. @param applicationContext Parameter containing information about the system @param partyId The party for which details to be fetched @return {@link PartyDetails} Object containing information about party @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyDetails(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method gives the age status i.e party is Minor/Major  @param applicationContext Parameter containing information about the system @param partyId The party for which age status need to be fetched @return {@link AgeStatus} Specifies that party is major or minor @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getAgeStatus(com.ofss.fc.framework.context.ApplicationContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This service will fetch all the existing parties that are holders for the term deposit account. @param sessionContext This parameter captures the basic information about the user of the system. @param accountId Term deposit account id @return {@link AccountHolderInquiryResponse} Object containing parties for TD account @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#fetchPartiesOfTdAccount(com.ofss.fc.app.context.SessionContext, java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#getPartyDetails(java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method returns if any of the party is black listed.  If the Blacklisting status of any of the holders is $\cap \neg \downarrow$ Blacklisted $\cap \neg \downarrow$ , Throw "TD account opening is not permitted Party is blacklisted".  @param partyId @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getPartyDetails	This method is used to fetch the details of party depending on the party id specified @param partyId The party for which party details need to be fetched @return {@link PartyInquiryResponse} Object containing party details	This method checks if the party is deceased.   It returns true if customer is $\cap \neg \downarrow$ Deceased $\cap \neg \downarrow$ else false.  @return boolean
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyDeceasedTD	This method is used to check the existence of deceased customer for a specific partyId @param partyId The party for which the flag PartyDeceasedTD need to be fetched @return boolean	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isPartyDeceasedTD	This method is used to check the existence of deceased customer for a specific partyId @param partyId The party for which the flag PartyDeceasedTD need to be fetched @return boolean	This method is used to check the existence of deceased customer for a specific partyId @param partyId The party for which the flag PartyDeceasedTD need to be fetched @return boolean @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#isPartyDeceasedTD(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isCustomerBlacklisted	This method is used to check the existence of blackListed customer for a specific partyId @param partyId The party for which the flag CustomerBlacklisted need to be fetched @return boolean	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isCustomerBlacklisted	This method is used to check the existence of blackListed customer for a specific partyId @param partyId The party for which the flag CustomerBlacklisted need to be fetched @return boolean	This method is used to check the existence of blackListed customer for a specific partyId @param partyId The party for which the flag CustomerBlacklisted need to be fetched @return boolean @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#isCustomerBlacklisted(java.lang.String)
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isKYCConfirmed	This method is responsible for calling the party module inorder to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li> <li>Else return false indication KYC is not done. @param partyId The party for which the flag KYCConfirmed need to be fetched @param roleId The role for which KYCConfirmed @return boolean @throws	This method is responsible for calling the party module inorder to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li> <li>Else return false indication KYC is not done. @param partyId The party for which the flag KYCConfirmed need to be fetched @param roleId The role for which KYCConfirmed @return boolean @throws FatalException @see com.ofss.fc.app.adapter.td.ITDPartyAdapter#isKYCConfirmed(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDPartyAdapter	isKYCConfirmed	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt; Else return false indication KYC is not done.</p> <p>@param partyId The party for which the flag KYCConfirmed need to be fetched @param roleId The role for which KYCConfirmed @return boolean @throws FatalException</p>	<p>This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: &lt;li&gt;Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default&lt;/li&gt; &lt;li&gt;This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output&lt;/li&gt; &lt;li&gt;if kycHistoryDTO.KYCStatus == Confirmed &amp;&amp; kycHistoryDTO.kycExpiryDate is greater than the posting date then return true &lt;/li&gt; Else return false indication KYC is not done. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	isKYCConfirmed	This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpirtyDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @param partyId The party for which the flag KYCConfirmed need to be fetched @param roleId The role for which KYCConfirmed @return boolean @throws FatalException	This method sets the account status based on the KYC status of the party  Setting the account status get the party KYC status. If it is confirmed then mark the account status as confirm else mark the account as AccountStatus.KYCCunconfirmed  If the KYC status of any of the holders is 'KYC Unconfirmed', open the TD account with status as 'Debit Block' with reason as 'KYC Unconfirmed'. @param applicationContext @param partyToAccountRelationshipDTOs
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllTermDepositAccountsOfParty	This service will fetch all existing party - TD account business relationship for the related party identifier passed @param sessionContext @param partyId @return PartyAccountAllRelationshipsInquiryResponse contains all party account relationships for the related party identifier passed @throws FatalException	This method will fetch the collateral details for the given party id. External service/module will call thisMethod to fetch the collateral values for all account and deposits by giving party ids and currencycode as input. @param sessionContext @param partyIds @param currencyCode @return TDCollateralResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllTermDepositAccountsOfParty	This service will fetch all existing party - TD account business relationship for the related party identifier passed @param sessionContext @param partyId @return PartyAccountAllRelationshipsInquiryResponse contains all party account relationships for the related party identifier passed @throws FatalException	This method will fetch the collateral liens placed on all Accounts for the given party id. External service/module will call this method to fetch the collateral values for all account and deposits by giving party ids as input. @param sessionContext @param partyIds @param collateralId @return TDCollateralResponse @throws FatalException
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchAllTermDepositAccountsOfParty	This service will fetch all existing party - TD account business relationship for the related party identifier passed @param sessionContext @param partyId @return PartyAccountAllRelationshipsInquiryResponse contains all party account relationships for the related party identifier passed @throws FatalException	
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyWarningIndicators	This method will provide the list of warning Indicators set for the given party id. @param partyId @return PartyWarningIndicatorInquiryResponse @throws FatalException	This method fetches the @param partyId @return PartyAppliedWarningIndicatorDTO[]

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyWarningIndicators	This method will provide the list of warning Indicators set for the given party id. @param partyId @return PartyWarningIndicatorInquiryResponse @throws FatalException	<pre>This method will validate the party id for any warning indicators to be present passed by the Origination module during account opening. There can be more than one party id in the PartyToAccountRelationship[] as there can be joint account opening request. For each party check the party Id has warning indicator. &lt;li&gt; Call {@link ITDPartyAdapter#fetchPartyWarningIndicators(sessionContext, partyId)}&lt;/li&gt; &lt;li&gt;&lt;ol&gt; isPartyExistWithRole can throw two exceptions &lt;li&gt; check if warning indicator is present on the party id&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; </pre>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		This method assembles the partyDetails from CustomerDetailsInquiryResponse @param customerDetailsInquiryResponse {@link CustomerDetailsInquiryResponse} Object having customer details which will be used to get data for assembling party details object @param partyDetails {@link Partydetails} Object which needs to be assembled
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	getAgeStatusBasedOnPartyMajorFlags		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyWarningIndicatorsBasedOnEventCode	This method fetches the warning indicators defined for a party,   and if these warning indicators are to be applied,   it fetches the actions for the passed eventId. @param eventId @param partyId @return @throws FatalException	<pre>This method will validate the party id for any warning indicators to be present passed by the Origination module during account opening. There can be more than one party id in the PartyToAccountRelationship[] as there can be joint account opening request. For each party check the party Id has warning indicator. &lt;li&gt; Call {@link ITDPartyAdapter#fetchPartyWarningIndicators(sessionContext, partyId)}&lt;/li&gt; &lt;li&gt;&lt;ol&gt; isPartyExistWithRole can throw two exceptions &lt;li&gt; check if warning indicator is present on the party id&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; </pre>
ITDPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDPartyAdapter	fetchPartyNamesOfTdAccount	This is a light API which fetches only some required party related information of an account the required information are partyId, Party Fullname. @param accountId @return @throws FatalException	This method populates the common fields required for alerts processing of all events  <li>Deposit related information</li> @param accountId @param depositId @return DepositEventDetailDTO
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	validateInterestRule		This method performs add operation. @param interestRuleDefinition
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	validateInterestRule		This method performs update operation. @param interestRuleDefinition

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	validateInterestRule		This method is used to do validations on the fields @param interestRuleDefinition
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	validateInterestRule		@param interestRuleId @param productClass @param interestClass @param interestDivision @param interestRuleVersion @param productCode @param tdAttributes
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	fetchInterestRule		This represents the fetchInterestRuleDefinition method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,InterestRuleDefinitionDTO,interestRuleDefinitionDTO, ,interestRuleDefinitionDTO TODO: Explain the interestRuleDefinitionDTO above @return {@link com.ofss.fc.app.pm.dto.interestruledefinition.InterestRuleDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IIInterestRuleDefinitionAdapter	fetchInterestRule		This represents the fetchByVersion method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,InterestRuleDefinitionDTO,interestRuleDefinitionDTO, ,interestRuleDefinitionDTO TODO: Explain the interestRuleDefinitionDTO above @return {@link DomainObjectDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IIInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IIInterestRuleDefinitionAdapter	fetchInterestRule		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		TODO:THIS SHOULD BE DROPPED @param productCode @param planCode @return
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		@param productCode @param planCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	fetchInterestRule		This checks whether the IRD linked to product has offset balance as Interest base component @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,productCode, ,productCode stores the product Code of the offer. @return {@link OfferInquiryServiceResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section Description: Method will fetch all the currency available for specific offer.
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	fetchInterestRule		@param loanOffer
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	fetchInterestRule		This method is used to check the existing interest rule and return the required details associated with the interest rule @param interestRuleKey which is combination of certain attributes @return Map
InterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\InterestRuleDefinitionAdapter	fetchInterestRule		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		This method fetches all the Interest Rules linked to a given product code @param sessionContext @param productCode @return
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRule		
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	isUnclearBalanceUsed		
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	isUnclearBalanceUsed		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	isOffsetBalanceSetForProduct	@param interestRuleDefinition @return	This checks whether the IRD linked to product has offset balance as Interest base component @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,productCode, ,productCode stores the product Code of the offer. @return {@link OfferInquiryServiceResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section Description: Method will fetch all the currency available for specific offer.
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRuleDTO	This method fetches all the Interest Rules linked to a given product code @param sessionContext @param productCode @return	@param loanOffer
IInterestRuleDefinitionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pm\IInterestRuleDefinitionAdapter	fetchInterestRuleLienAllowed		TODO:THIS SHOULD BE DROPPED @param productCode @param planCode @return
IOIMIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	addUser		This method is used to add user in OIM with all the attributes set in Oracle Identity DTO.  @param oracleIdentityDTO @throws FatalException
IOIMIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	addUser		This method adds the user to the groups in the OIM.  @param userId @param rolist @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	addUser		<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus}&lt;br&gt; @throws com.ofss.fc.infra.exception.FatalException&lt;br&gt; Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	addUser		<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus}&lt;br&gt; @throws com.ofss.fc.infra.exception.FatalException&lt;br&gt; Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	updateUser	This method updates the user in the OIM.  @param oracleIdentityDTO @throws FatalException	This method updates the user in the OIM.  @param oracleIdentityDTO @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	updateUser	This method updates the user in the OIM.  @param oracleIdentityDTO @throws FatalException	This represents the updateUser method of the entity.  It updates the attributes of the user in the OIM.  It does not update the password of the user.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	Fetch user profile using any attribute name as criteria @param criteria @param criteriaValue @return IdentityIxfceDTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	This represents the inquireUserWithCriteria method of the entity. This method fetches the details of the user based on some criteria.  Following two criteria are allowed for the user inquiry :  <ul> <li>1. User Login</li> <li>2. Email</li> <li>3. USERS_PARTY_ID</li> </ul> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.  @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUser	Fetch user with user id as input key parameter @param userId @return IdentityIxfceDTO @throws FatalException	This represents the inquireUsers method of the entity. This method fetches information about all the Users that were created on that particular Day  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUserWithCriteria	Fetch user profile using any attribute name as criteria @param criteria @param criteriaValue @return IdentityIxfaceDTO @throws FatalException	Fetch user profile using any attribute name as criteria @param criteria @param criteriaValue @return IdentityIxfaceDTO @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUserWithCriteria	Fetch user profile using any attribute name as criteria @param criteria @param criteriaValue @return IdentityIxfaceDTO @throws FatalException	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	inquireUserWithCriteria	Fetch user profile using any attribute name as criteria @param criteria @param criteriaValue @return IdentityIxfceDTO @throws FatalException	This represents the inquireUserWithCriteria method of the entity. This method fetches the details of the user based on some criteria.  Following two criteria are allowed for the user inquiry :  <ul> <li>1. User Login</li> <li>2. Email</li> <li>3. USERS_PARTY_ID</li> </ul> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.  @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	changeEmailId	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newEmailId @throws FatalException	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newEmailId @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	changeEmailId	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newEmailId @throws FatalException	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newUserId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	changeEmailId	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newEmailId @throws FatalException	This represents the changeLoginEmailId method of the entity.  It changes the email id of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	changeUserId	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newUserId @throws FatalException	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newUserId @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	changeUserId	Change the login id of the existing user in OIM. OIM autocheck the duplicate user Id @param userId @param newUserId @throws FatalException	This represents the changeUserId method of the entity. This method changes the user Id of the user.  The userId is the current user Id and newUserId is the changed user id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @fcb.param MI,String,newUserId, ,It represents the new User id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	lockAccount	lock the user account with user id as parameter @param userId @throws FatalException	lock the user account with user id as parameter @param userId @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	lockAccount	lock the user account with user id as parameter @param userId @throws FatalException	This represents the lockAccount method of the entity. This method locks the account of the user in OIM.  If the account of the user is locked, then the user won't be able to login to his account @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,userId, It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	unlockAccount	unlock the already lock account with user id as key parameter @param userId @throws FatalException	unlock the already lock account with user id as key parameter @param userId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	unlockAccount	unlock the already lock account with user id as key parameter @param userId @throws FatalException	This represents the unlockAccount method of the entity. This method unlocks the account of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus}  @throws com.ofss.fc.infra.exception.FatalException  Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setOrResetPasswordChangeAtNextLogonFlag	It indicates whether the password is to be changed at next logon.  @param userId @param flag @throws FatalException	It indicates whether the password is to be changed at next logon.  @param userId @param flag @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	It indicates whether the password is to be changed at next logon.  @param userId @param flag @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	This method resets the password of the user. The OIM admin logs into the OIM  and resets the user password. @param userId @param password @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	This method validates the user password.  If the password is null or an empty string, it sets the default password.  @param identityDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	This represents the resetPassword method of the entity. This method resets the password of the user.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	setPassword	This method changes the password of the user in OIM.  @param userId @param oldPassword @param changedPassword @throws FatalException	This represents the setPassword method of the entity. This method sets the password of the user.  Here the currentPassword is the current password of the user and  and the newPassword is the changed password.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @fcb.param MI,String,currentPassword, ,It represents the current password of the user.  @fcb.param MI,String,newPassword, ,It represents the new password of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	resetPassword	This method resets the password of the user. The OIM admin logs into the OIM  and resets the user password. @param userId @param password @throws FatalException	This method resets the password of the user. The OIM admin logs into the OIM  and resets the user password. @param userId @param password @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	resetPassword	This method resets the password of the user. The OIM admin logs into the OIM  and resets the user password. @param userId @param password @throws FatalException	This represents the resetPassword method of the entity. This method resets the password of the user.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	addUserTo Group	This method adds the user to the groups in the OIM.  @param userId @param rolelist @throws FatalException	This method adds the user to the groups in the OIM.  @param userId @param rolelist @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	revokeRole	This method removes the user from a group in the OIM.  @param userId @param roleList @throws FatalException	This method removes the user from a group in the OIM.  @param userId @param roleList @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	deleteUser	This method deletes a user from OIM.  @param userId @throws FatalException	This method deletes a user from OIM.  @param userId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	maintainUsers	This method manages multiple users in OIM.  @param identityIxfceDTOs @throws FatalException	This method manages multiple users in OIM.  @param identityIxfceDTOs @throws FatalException
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	maintainUsers	This method manages multiple users in OIM.  @param identityIxfceDTOs @throws FatalException	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	fetchRoles	This method fetch roles assigned to users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTO	This method fetch roles assigned to users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTO
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	fetchRoles	This method fetch roles assigned to users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTO	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	fetchRoles	This method fetch roles assigned to users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTO	
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	searchUsers	This method manages multiple users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTOs	This method manages multiple users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTOs
IOIMIdentity Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOIMIdentityAdapter	searchUsers	This method manages multiple users in OIM.  @param userId @throws FatalException @return IdentityIxfceDTOs	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyWarningIndicatorAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IPartyWarningIndicatorAdapter	publishPartyWarningIndicator		
IInterestLoanApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\interest\IInterestLoanApplicationAdapter	fetchLoanAccountStatus	This @param accountId @return	This method validates whether the interest adjustment is restricted or not based on the account status. <ol> <li>If Product Class is {@link ProductClass#SAVINGS} or {@link ProductClass#CHECKING} </li> <li>Derive the adjustment {@link Event} using {@link EventResolutionService#deriveEvent(ProductClass, InterestClass, InterestDivision, InterestConstituentType, String, AccountingAmountType, boolean)} </li> <li>Prepare the {@link InterestRestrictionRequest} using the event derived in 1.1 and {@link InterestPlanItem#getInterestAgeingStatus()}</li> <li>Set {@link InterestRestrictionRequest#getTransactionType()} as {@link TransactionType#Credit} if {@link InterestAdjustmentRequestDTO#getEntryType()} is equal to {@link AdjustmentAmountType#INCREASE}, else set it to {@link TransactionType#Debit}. </li> <li>Call {@link InterestRestrictionAdapter#isServiceRestricted(InterestRestrictionRequest)} and receive the response isServiceRestricted</li> <li>If isServiceRestricted == true , throw business exception with error code : {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED}</li> </ol> </li> <li>Else if Product Class is {@link ProductClass#TERM_DEPOSITS} </li> <li>Call

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link InterestRestrictionAdapter#isAdjustmentAllowedForDeposit(String, int)} and receive the response isAllowed.&lt;/li&gt; &lt;li&gt;If isAllowed == false, throw business exception with error code : {@link InterestErrorConstants#INT_ERROR_ADJUSTMENT_NOT_ALLOWED_FOR_DEPOSIT}&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt;&lt;/ol&gt; @param interestAdjustmentRequestDTO @param interestPlanItem @param accountId @param depositId @param interestRule</p>
ICardOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardOriginationAdapter	createOrUpdateAccountReferenceDetails	This method is used to create or update account reference details @param sessionContext @param creditCardAccountConfigurationDTO @param applicationId @param productGroupSerialNumber @return @throws FatalException	This method is used to create or update account reference details @param sessionContext @param creditCardAccountConfigurationDTO @param applicationId @param productGroupSerialNumber @return @throws FatalException
ICardOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardOriginationAdapter	createOrUpdateAccountReferenceDetails	This method is used to create or update account reference details @param sessionContext @param creditCardAccountConfigurationDTO @param applicationId @param productGroupSerialNumber @return @throws FatalException	The method createOrUpdateCreditCardAccountDetails is exposed for external system to create or update credit card account in OBP in async flow  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardOriginationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardOriginationAdapter	createOrUpdateAccountReferenceDetails	This method is used to create or update account reference details @param sessionContext @param creditCardAccountConfigurationDTO @param applicationId @param productGroupSerialNumber @return @throws FatalException	
IDDANPAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAAdapter	notifyForNewAccount	This method will call Account Asset Classification Helper while account opening to get resister in NPA module. @param accountId @return accountClassificationResponse	This method calls {@linkAccountAssetClassificationHelper#notifyForNewAccount} while new account opening to get resister in NPA module. @param accountId @return accountClassificationResponse
IDDANPAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAAdapter	notifyForNewAccount	This method will call Account Asset Classification Helper while account opening to get resister in NPA module. @param accountId @return accountClassificationResponse	
IDDANPAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAAdapter	notifyForNewAccount	This method will call Account Asset Classification Helper while account opening to get resister in NPA module. @param accountId @return accountClassificationResponse	This method will call NPA module through adapter call and gives CRR code and Accrual Status Type in response. @param accountId @return accountClassificationResponse @throws FatalException
IDDANPAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAAdapter	notifyForNewAccount	This method will call Account Asset Classification Helper while account opening to get resister in NPA module. @param accountId @return accountClassificationResponse	This method will call NPA module through adapter call passing the back dated account opening date as input and gives CRR code and Accrual Status Type in response. @param accountId @return accountClassificationResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyForAccountUpdate	This method will call Account Asset Classification Helper while OD facility adding or removing over account. @param accountId @param facilityId @return accountClassificationResponse	This method calls NPA module through adapter which will give response as updated CRR code and it gets stamp in demand deposit account. @param accountId @param facilityId @return accountClassificationResponse @throws FatalException
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyForAccountUpdate	This method will call Account Asset Classification Helper while OD facility adding or removing over account. @param accountId @param facilityId @return accountClassificationResponse	This method calls {@link AccountAssetClassificationHelper#notifyForAccountUpdate} while OD facility adding or removing over account. @param accountId @param facilityId @return accountClassificationResponse
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyForAccountUpdate	This method will call Account Asset Classification Helper while OD facility adding or removing over account. @param accountId @param facilityId @return accountClassificationResponse	This method calls NPA module through adapter which will give response as updated CRR code and it gets stamp in demand deposit account. @param accountId @param facilityId @return accountClassificationResponse @throws FatalException
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyForAccountUpdate	This method will call Account Asset Classification Helper while OD facility adding or removing over account. @param accountId @param facilityId @return accountClassificationResponse	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	freezeDPD HardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is 07. Freeze DPD Calculation 07. then it will call the AccountAssetClassificationService to freeze the DPD calculation. @param accountId @param reliefStartDate @param reliefEpiryDate	This method will be called from hardShipReliefExecution if hardShipReliefType is 07. Freeze DPD Calculation 07. then it will call the AccountAssetClassificationService to freeze the DPD calculation. @param accountId @param reliefStartDate @param reliefEpiryDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	freezeDPD HardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is Freeze DPD Calculation then it will call the AccountAssetClassificationService to freeze the DPD calculation. @param accountId @param reliefStartDate @param reliefEpiryDate	This method will be called from hardShipReliefExecution if hardShipReliefType is "Freeze Interest" and interest rate type is "fixed" then it will call Interest module for HardShip Relief action for freezing interest. It will call the DDAPricingAdapter to call the InterestFreezeService to freeze the interest if overdraft account has fixed rate. @param applicationContext @param hardShipReliefDetails @throws FatalException
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	freezeDPD HardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is Freeze DPD Calculation then it will call the AccountAssetClassificationService to freeze the DPD calculation. @param accountId @param reliefStartDate @param reliefEpiryDate	This method will be called from hardShipReliefExecution if hardShipReliefType is "Freeze DPD Calculation" then it will call Asset Classification adapter for HardShip Relief action for freezing DPD. It will call the AccountAssetClassificationService to freeze the DPD calculation. @param applicationContext @param hardShipReliefDetails
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	freezeDPD HardShipRelief	This method will be called from hardShipReliefExecution if hardShipReliefType is Freeze DPD Calculation then it will call the AccountAssetClassificationService to freeze the DPD calculation. @param accountId @param reliefStartDate @param reliefEpiryDate	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	getProvisionAmount	This method provide the Provision Amount for the given account @param accountId @return	This is an internal service to prepare accounting leg @param applicationContext @param amountTag @param txnAmount @param ddaAccount @param isAccountCredit @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	getProvisionAmount	This method provide the Provision Amount for the given account @param accountId @return	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	updateProvisionDrawdownAmount	This method update the Provision Amount for the given account by the provided amount. @param accountId @param provisionDrawdownAmount	This is an internal service to prepare accounting leg @param applicationContext @param amountTag @param txnAmount @param ddaAccount @param isAccountCredit @return
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	updateProvisionDrawdownAmount	This method update the Provision Amount for the given account by the provided amount. @param accountId @param provisionDrawdownAmount	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	The method will delink OD facility from account. It does the following: <ol> <li>Set the facilityId, startDate in {@link DemandDepositAccount} to null</li> <li>If the isODInFixedRatePeriod is true, then set it to false and if this action coming from EOD due to Unclaimed processing or any other case then it logs the rate change action on next day BOD otherwise If it is part of BOD then it directly executes the action there itself.</li> <li>If the account has any offset accounts, close all the offset linkages</li> <li>Call {@link DemandDepositAccount#reapportionFunds()}</li> <li>Check if the facility is active. If yes then intimate eLCM on od linkage expiry</li> <li>Balance change action will be logged by reapportionFunds() method.</li> </ol> @param accountId @param facilityId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	This service updates the CRR code at account level . At the time of update this service calls eligibility framework to find account level access changes and same time update the flowing account access parameters AUTO_REORDER_CHEQUEBOOK ,ACCOUNT_EXCESS ,PRINCIPAL_OFFSET, DEBIT_BLOCK @param applicationContext @param accountId @param crrCode
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	This method would fetch the session context. @param applicationContext @param sessionContext @return
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	This method is invoked after the account writeoff operation iff MDM integration is enabled. This method fetched {@link IMDMBatchIntegrationAdapter} and populates {@link IMDMBatchIntegrationDTO}  and invokes logging into FLX_INTEGRATE_MDM_BATCH_LOG table, where is it picked for publish to MDM. @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				applicationContext @param accountId
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchSessionContext	This method will provide the session context out of the application context @param accountId @return	This method fetches adapter for logging account information during batch execution. @param applicationContext @param demandDepositAccount
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchDPD	This method will be called from DDAAccountCollectionIntimationApplication to fetch DPD for an account. @param accountId @return dpd	This method will be called from DDAAccountCollectionIntimationApplication to fetch DPD for an account. @param accountId @return dpd
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchDPD	This method will be called from DDAAccountCollectionIntimationApplication to fetch DPD for an account. @param accountId @return dpd	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	fetchDPD	This method will be called from DDAAccountCollectionIntimationApplication to fetch DPD for an account. @param accountId @return dpd	This method returns integer value DPD for an account. @param accountId
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyOnStatusChange	This method will be called from account closure and write off. @param accountId @param statusCode	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyOnStatusChange	This method will be called from account closure and write off. @param accountId @param statusCode	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyOnStatusChange	This method will be called from account closure and write off. @param accountId @param statusCode	This method is common activity which can be called by any service for activities such as 1.validate & delink bundle benefits from the account to be closed by call to bundle 2.close pricing plan by calling pricing 3.close interest plan by calling interest 4.unregister taxation plan by calling taxation 5.remove all the actions on the account 6.Remove all proportioned OD logged from effective date. 7.delink alternate account. @param closureDTO @param account @param agingAccountStatus @param finalBalance @throws FatalException
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDANPAAadapter	notifyOnStatusChange	This method will be called from account closure and write off. @param accountId @param statusCode	This method is called by the batch process to change the aging account status. If an action to change the aging account status from one status to another. For instance, AgingAccountStatus is always changed in the sequence. AgingAccountStatus#Regular -> AgingAccountStatus#Inactive AgingAccountStatus#Inactive -> AgingAccountStatus#Dormant AgingAccountStatus#Dormant -> AgingAccountStatus#Unclaimed Each will be set by the corresponding private method. For a account on the given actionDate, only one of these private method can be called.  Any other combination cannot be applied. An account which is once marked as AgingAccountStatus#Unclaimed, is not allowed to reset its AgingAccountStatus. TODO : When an account is marked unclaimed, accounting entries are also passed. This method works as follows :  <pre> accountId = actionSet.getKey().getAccountId() actionDate = actionSet.getKey().getActionDate() 1. Load DemandDepositAccount from repository. 2. If(currentAgingDetails.getInactiveDate() == actionDate) Call setInactiveAgingAccountStatus. 3. Else If(currentAgingDetails.getDormantDate() ==

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				actionDate) Call setDormantAgingAccountStatus. 4. Else If(currentAgingDetails.getUnclaimedDate() == actionDate) Call setUnclaimedAgingAccountStatus. 5. Update DemandDepositAccount calling repository. &#064;param applicationContext &#064;param accountBasedActionKey &#064;param actionRequest @param dueDate
IDDANPAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dda\IDDANPAAapte r	creditFunds OnClosedO rWrittenOff Accounts	Credit payments are allowed on asset type of accounts which are closed or written off. Following steps are performed . <ol> <li> Product maintains arrear appropriation sequence for credit payments on written off amount. <li> On credit payments on closed or written off accounts, demand posit module will not perform any action. It will redirect call to NPA module . NPA module will recover arrear based on arrear appropriation sequence. <li> NPA module will give accounting tags which will be given to payments module. </ol> @param sessionContext @param accountTransactionDetailsDTO @param productId @param branchId @param offerId @param partyId @return AccountingEventAmountTagDetails DTO[],accounting tags from NPA module.	This method is used to credit account identified by accountTransaction.accountId. The financial method works as follows: <ol> <li>Call {@link getInternalTransactionReferenceNumber(): internalTransactionReferenceNumber} to get the internalTransactionReferenceNumber</li> <li>validate existence of the transaction by super.validateDuplicateTransaction(internalTransactio nReferenceNumber: String, SessionContext.externalReferenceNumber: String)</li> <li>Convert the AccountTransactionDetailsDTO to AccountTransaction using {@link AccountTransactionDetailsAssembler#toInformationO bject (accountTransactionDetailsDTO: AccountTransactionDetailsDTO): : AccountTransaction} <li>Set FinancialEvent as: accountTransaction.FinancialEventType=FinancialEve ntType.Regular</li> <li>Set eventId as: accountTransaction.TransactionAttributes.EventId="C ASA001"</li> <li>Set internalTransactionReferenceNumber as: accountTransaction.TransactionReferenceNumber=int ernalTransactionReferenceNumber</li> <li>Check business policies by calling {@link checkCreditAccountPolicies(accountTransaction: AccountTransaction)} <li>If account is closed or

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>written off,credit payments are allowed.Call is redirected to <a href="#">{@link #creditFundsOnClosedOrWrittenOffAccounts(Session Context, AccountTransactionDetailsDTO, IDemandDepositAccount)}</a> Which notify NPA(Asset classification) module. <ul style="list-style-type: none"> The business policy for the credit account needs to be qualified. Please refer to each method for explanation of the validation. <li><a href="#">{@link CreditTransactionBusinessPolicy#validateAccount()}&lt;/li&gt;&lt;/li&gt; <a href="#">{@link CreditTransactionBusinessPolicy#validateTransactionAmount()}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;convert the Transaction.AccountTransaction.TransactionAmounts.accountAmount into transactionAmount by calling super.convertToCurrencyAmount(transactionAmount s: TransactionAmounts,sourceCurrency:SourceCurrency )&lt;/li&gt; &lt;li&gt;call super.manageTransactionDates(transactionDates: TransactionDates, transactionBranch: TransactionBranch) for posting date validation.&lt;/li&gt; &lt;li&gt;call <a href="#">{@link DemandDepositAccountService#creditFunds(account Transaction: AccountTransaction): CreditResponse}</a> for crediting the account with the transaction amount.&lt;/li&gt; &lt;li&gt;The following domain service will be called for calculating and charging the fees.&lt;br&gt;<a href="#">{@link super.manageFees(accountTransaction: AccountTransaction, feeDetails: FeeDetails, ModuleType.CASA)}</a>&lt;/li&gt; &lt;li&gt;Override conditions <ul style="list-style-type: none"> There is another business policy to check if the account or the account holder falls in the override conditions. &lt;li&gt;if customer is blacklisted then it is also comes under the override condition&lt;/li&gt; &lt;li&gt;if there</ul></a></a></li></ul></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>is high/medium severity memo then this is also a override condition&lt;/li&gt; &lt;li&gt;Sms module will have configuration if to ignore/authorize/reject the transaction.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ol&gt; @param SessionContext MI sessionContext: This parameter captures the basic information about the user of the system. &lt;br&gt; The information is used by the system for authentication purposes. @param AccountTransactionDetailsDTO MI accountTransactionDetailsDTO : This parameter carries the basic information about the transaction. @return @throws FatalException</p>
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	creditFunds OnClosedOrWrittenOff Accounts	<p>Credit payments are allowed on asset type of accounts which are closed or written off. Following steps are performed . &lt;ol&gt; &lt;li&gt; Product maintains arrear appropriation sequence for credit payments on written off amount. &lt;li&gt; On credit payments on closed or written off accounts, demand posit module will not perform any action. It will redirect call to NPA module . NPA module will recover arrear based on arrear appropriation sequence. &lt;li&gt; NPA module will give accounting tags which will be given to payments module. &lt;/ol&gt; @param sessionContext @param accountTransactionDetailsDTO @param productId @param branchId @param offerId @param partyId @return AccountingEventAmountTagDetails</p>	<p>Credit payments are allowed on asset type of accounts which are closed or written off. Following steps are performed . &lt;ol&gt; &lt;li&gt;Product maintains arrear appropriation sequence for credit payments on written off amount. &lt;li&gt;On credit payments on closed or written off accounts, demand posit module will not perform any action. It will redirect call to NPA module . NPA module will recover arrear based on arrear appropriation sequence. &lt;li&gt;NPA module will give accounting tags which will be given to payments module. &lt;li&gt; &lt;/ol&gt; @param sessionContext @param accountTransactionDetailsDTO @param accountTransaction @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			DTO[],accounting tags from NPA module.	
IDDANPAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDANPAAadapter	creditFunds OnClosedOrWrittenOff Accounts	<p>Credit payments are allowed on asset type of accounts which are closed or written off. Following steps are performed .</p> <ul style="list-style-type: none"> <li>&lt;li&gt; Product maintains arrear appropriation sequence for credit payments on written off amount.</li> <li>&lt;li&gt; On credit payments on closed or written off accounts, demand posit module will not perform any action. It will redirect call to NPA module . NPA module will recover arrear based on arrear appropriation sequence.</li> <li>&lt;li&gt; NPA module will give accounting tags which will be given to payments module.</li> </ul> <p>@param sessionContext @param accountTransactionDetailsDTO @param productId @param branchId @param offerId @param partyId @return AccountingEventAmountTagDetails DTO[],accounting tags from NPA module.</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data.  @param sessionContext : {@link SessionContext} @param bankDTO : {@link BankDTO} @return : {@link BankInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param stateDTO : {@link StateDTO} @return : {@link StateInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link BranchParametersInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param paymentTypeCodesMaintDTO :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse that is imported from the exported location @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param reasonCodesDTO :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen.  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTO : {@link BankPolicyDefinitionDTO} @return : {@link BankPolicyDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext : {@link SessionContext} @param verificationCategoryDto : {@link VerificationCategoryDto} @return : {@link VerificationCategoryInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext : {@link SessionContext} @param countryCodeDTO : {@link CountryCodeDTO} @return : {@link CountryCodeInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext : {@link SessionContext} @param finYearDefinitionDTO : {@link FinYearDefinitionDTO} @return : {@link FinYearDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This method represents importAndInquireAllStatusRestrictions of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse}   AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.  <b>accountStatus</b> account status  <b>accountStatusType</b> account status type  <b>serviceDTOs</b> array of serviceDTO which will contain -   <i>id :</i> ServiceId  <i>name :</i> Name of Service  <i>description :</i> Description of restricted service. 
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param bankParametersDTO :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param purposeDTO :{@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	importBusinessEntity	This method imports the response object of the Business Entity for the provided key, task code.  The response object, containing the DTO, is the latest exported object.  For example, if a business entity has been exported multiple times, this method will import the latest response. @param SessionContext @param BaseResponse @param String key, String service @throws FatalException @returns BaseResponse	This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	exportAllBusinessEntities	This method returns the config package ID for all the exported business entities for a task code.  The config package ID is encapsulated in the BaseResponse object.   	This method calls ConfigDataMoveAdapter.fetchAllAndExport with sessionContext,taskCode as input  @param sessionContext @param taskCode @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param bankDTO : {@link BankDTO} @return : {@link BankInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This method takes list of DTO as input and fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method. @param sessionContext : {@link SessionContext} @param bankDTOs : Array of {@link BankDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param stateDTO : {@link StateDTO} @return : {@link StateInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param stateDTOs : Array of {@link StateDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link BranchParametersInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.   @param sessionContext : {@link SessionContext} @param branchParametersDTOs : Array of {@link BranchParametersDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param paymentTypeCodesMaintDTO :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param paymentTypeCodesDTOs :Array of {@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity and then calls the adapter for further processing It calls the exportBusinessEntity method of the entity with the appropriate request     @fcb.param ,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param reasonCodesDTO :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param reasonCodesDTOs :An array of {@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location.  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTO : {@link BankPolicyDefinitionDTO} @return : {@link BankPolicyDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTOs : Array of {@link BankPolicyDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param verificationCategoryDTO : {@link VerificationCategoryDTO} @return : {@link VerificationCategoryInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param verificationCategoryDTOs : Array of {@link VerificationCategoryDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param countryCodeDTO : {@link CountryCodeDTO} @return : {@link CountryCodeInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param countryCodeDTOs : Array of {@link CountryCodeDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param finYearDefinitionDTO : {@link FinYearDefinitionDTO} @return : {@link FinYearDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param finYearDefinitionDTOs : Array of {@link FinYearDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This method represents inquireAllStatusRestrictionsAndExport of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse}   AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.  <b>accountStatus</b> account status  <b>accountStatusType</b> account status type  <b>serviceDTOs</b> array of serviceDTO which will contain -   <i>id :</i> ServiceId  <i>name :</i> Name of Service  <i>description :</i> Description of restricted service. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This method represents inquireAllStatusRestrictionsAndExport of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse}   AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.  <b>accountStatus</b> account status  <b>accountStatusType</b> account status type  <b>serviceDTOs</b> array of serviceDTO which will contain -   <i>id :</i> ServiceId  <i>name :</i> Name of Service  <i>description :</i> Description of restricted service. 
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param bankParametersDTO :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  @param bankParametersDefinitionDTOs :Array of :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param purposeDTO :{@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}  This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param purposeDTOs :An Array of {@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param verificationPolicyDTOs : Array of {@link VerificationPolicyDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	generateID	This method returns the unique configpackageId for a BusinessEntity	This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	importAllBusinessEntities	This method imports all the BusinessEntities for a given taskCode and given ConfigPackageId @param context @param taskCode @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	exportAllSeedEntities	This method exports all the seed entities @param context @param dynamicEntityResponse @param service @throws FatalException	
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext : {@link SessionContext} @param bankDTO : {@link BankDTO} @return : {@link BankInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This method will get called from ExportImportApplicationService to export all records for the selected taskCode.  This method load all the entities for the input taskCode and make the domainObjectDTO from the Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response.  The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities method of ExportImportApplicationService. @param sessionContext : {@link SessionContext} @param entityName : String @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This method takes list of DTO as input and fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method. @param sessionContext : {@link SessionContext} @param bankDTOs : Array of {@link BankDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param stateDTO : {@link StateDTO} @return : {@link StateInquiryResponse} @throws FatalException : If any error occurs, {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : {@link String} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param stateDTOs : Array of {@link StateDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This method calls ConfigDataMoveAdapter.fetchAllAndExport with sessionContext,taskCode as input  @param sessionContext @param taskCode @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link BranchParametersInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : String @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param branchParametersDTOs : Array of {@link BranchParametersDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param paymentTypeCodesMaintDTO :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode &lt;br/&gt; &lt;br/&gt;This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities&lt;br/&gt; method of ExportImportApplicationService. &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param entityName:{@link java.lang.String} entity name of the taskCode. @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param paymentTypeCodesDTOs :Array of {@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity and then calls the adapter for further processing It calls the exportBusinessEntity method of the entity with the appropriate request     @fcb.param ,com.ofss.fc.app.context.SessionContext,sessionContext, ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input and fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param reasonCodesDTO :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode &lt;br/&gt; &lt;br/&gt;This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities&lt;br/&gt; method of ExportImportApplicationService. &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param entityName :{@link java.lang.String}, entity name of the taskCode @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param reasonCodesDTOs :An array of {@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location.  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTO : {@link BankPolicyDefinitionDTO} @return : {@link BankPolicyDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode  This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities method of ExportImportApplicationService. @param sessionContext : {@link SessionContext} @param entityName : String @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTOs : Array of {@link BankPolicyDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param verificationCategoryDTO : {@link VerificationCategoryDTO} @return : {@link VerificationCategoryInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : {@link String} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.   @param sessionContext : {@link SessionContext} @param verificationCategoryDTOs : Array of {@link VerificationCategoryDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param countryCodeDTO : {@link CountryCodeDTO} @return : {@link CountryCodeInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : {@link String} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param countryCodeDTOs : Array of {@link CountryCodeDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputted taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param finYearDefinitionDTO : {@link FinYearDefinitionDTO} @return : {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FinYearDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : String @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param finYearDefinitionDTOs : Array of {@link FinYearDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  List&lt;EntityDTO&gt; entityDTOList  <b>@return</b>  BaseResponse  <b>@throws</b> FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode</p> <p>This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetchAndExport method of the service to export the record</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  java.lang.String entity name of the taskCode  <b>@return</b>  entityInquiryResponse  <b>@throws</b> FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param bankParametersDTO :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode &lt;br/&gt; &lt;br/&gt;This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities&lt;br/&gt; method of ExportImportApplicationService. &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param entityName :{@link java.lang.String} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; @param bankParametersDefinitionDTOs :Array of :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param purposeDTO :{@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode &lt;br/&gt; &lt;br/&gt;This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities&lt;br/&gt; method of ExportImportApplicationService. &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param entityName :{@link java.lang.String} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param purposeDTOs :An Array of {@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				:{@link com.ofss.fc.infra.exception.FatalException}
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @param sessionContext : {@link SessionContext} @param entityName : {@link String} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param verificationPolicyDTOs : Array of {@link VerificationPolicyDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI.java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		This represents the fetchAllAndExport method for the taskCode. This method will get called from ExportImportApplicationService to export all records for the selected taskCode    This method load all the entity for the inputed taskCode and make the domainObjectDTO from the specific Assembler. For each record from the loaded entity list it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File in the exportAllBusinessEntities  method of ExportImportApplicationService.   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param MI,java.lang.String entity name of the taskCode @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	export		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param bankDTO : {@link BankDTO} @return : {@link BankInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data.    @param sessionContext : {@link SessionContext} @param bankDTO : {@link BankDTO} @return : {@link BankInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method aims at importing all the exported data for the entity and updating the same in the target environment. This method first fetches the record in the target environment. If it persists then it updates the same with the input response else makes an add operation. @param sessionContext : {@link SessionContext} @param response : {@link BankInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method takes list of DTO as input and fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method. @param sessionContext : {@link SessionContext} @param bankDTOs : Array of {@link BankDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  List&lt;EntityDTO&gt; entityDTOList  <b>@return</b>  BaseResponse  <b>@throws</b> FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@param</b> sessionContext : {@link SessionContext}  <b>@param</b> stateDTO : {@link StateDTO}  <b>@return</b> : {@link StateInquiryResponse}  <b>@throws</b> FatalException : If any error occurs, {@link FatalException} is thrown.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param stateDTO : {@link StateDTO} @return : {@link StateInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link StateInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param stateDTOs : Array of {@link StateDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link BranchParametersInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param branchParametersDTO : {@link BranchParametersDTO} @return : {@link BranchParametersInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link BranchParametersInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param branchParametersDTOs : Array of {@link BranchParametersDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param paymentTypeCodesMaintDTO :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen &lt;br/&gt;This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param paymentTypeCodesMaintDTO :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment &lt;br/&gt;This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation . &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param response :{@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintInquiryResponse} @return :{@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param paymentTypeCodesDTOs :Array of {@link com.ofss.fc.app.commonservice.core.dto.payment.PaymentTypeCodesMaintDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity and then calls the adapter for further processing It calls the exportBusinessEntity method of the entity with the appropriate request     @fcb.param ,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse that is imported from the exported location @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input and fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@param</b>  List&lt;EntityDTO&gt; entityDTOList</p> <p><b>@return</b>  BaseResponse</p> <p><b>@throws</b> FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location.</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@param</b>  sessionContext :{@link  com.ofss.fc.app.context.SessionContext}</p> <p>This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p><b>@param</b>  reasonCodesDTO :{@link  com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO}</p> <p><b>@return</b> :{@link  com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse}</p> <p><b>@throws</b> :{@link  com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen &lt;br/&gt;This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param reasonCodesDTO :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment &lt;br/&gt;This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation . &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param response :{@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesInquiryResponse} @return :{@link com.ofss.fc.service.response.TransactionStatus} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param reasonCodesDTOs :An array of {@link com.ofss.fc.app.commonservice.core.dto.reasoncode.ReasonCodesDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location.  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTO : {@link BankPolicyDefinitionDTO} @return : {@link BankPolicyDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen.  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTO : {@link BankPolicyDefinitionDTO} @return : {@link BankPolicyDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link BankPolicyDefinitionInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param bankPolicyDefinitionDTOs : Array of {@link BankPolicyDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method will be used to dump the calendar change details and intimate all the subscribers to propagate the changes. @param sessionContext @param dd @param WOrH @param calendarType @param branchCode @param maintenanceType
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param verificationCategoryDTO : {@link VerificationCategoryDTO} @return : {@link VerificationCategoryInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param verificationCategoryDTO : {@link VerificationCategoryDTO} @return : {@link VerificationCategoryInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link VerificationCategoryInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param verificationCategoryDTOs : Array of {@link VerificationCategoryDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>&lt;br/&gt; &lt;br/&gt; @param sessionContext : {@link SessionContext} @param countryCodeDTO : {@link CountryCodeDTO} @return : {@link CountryCodeInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param countryCodeDTO : {@link CountryCodeDTO} @return : {@link CountryCodeInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link CountryCodeInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.   @param sessionContext : {@link SessionContext} @param countryCodeDTOs : Array of {@link CountryCodeDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @param sessionContext : {@link SessionContext} @param finYearDefinitionDTO : {@link FinYearDefinitionDTO} @return : {@link FinYearDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialized response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @param sessionContext : {@link SessionContext} @param finYearDefinitionDTO : {@link FinYearDefinitionDTO} @return : {@link FinYearDefinitionInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link FinYearDefinitionInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param finYearDefinitionDTOs : Array of {@link FinYearDefinitionDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This method represents importAndInquireAllStatusRestrictions of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse} &lt;br&gt; AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.&lt;br&gt; &lt;b&gt;accountStatus&lt;/b&gt; account status&lt;br&gt; &lt;b&gt;accountStatusType&lt;/b&gt; account status type&lt;br&gt; &lt;b&gt;serviceDTOs&lt;/b&gt; array of serviceDTO which will contain - &lt;br&gt; &lt;i&gt;id :&lt;/i&gt; ServiceId&lt;br&gt; &lt;i&gt;name :&lt;/i&gt; Name of Service&lt;br&gt; &lt;i&gt;description :&lt;/i&gt; Description of restricted service.&lt;br&gt;</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This method represents inquireAllStatusRestrictionsAndExport of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse} &lt;br&gt; AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.&lt;br&gt; &lt;b&gt;accountStatus&lt;/b&gt; account status&lt;br&gt; &lt;b&gt;accountStatusType&lt;/b&gt; account status type&lt;br&gt; &lt;b&gt;serviceDTOs&lt;/b&gt; array of serviceDTO which will contain - &lt;br&gt; &lt;i&gt;id :&lt;/i&gt; ServiceId&lt;br&gt; &lt;i&gt;name :&lt;/i&gt; Name of Service&lt;br&gt; &lt;i&gt;description :&lt;/i&gt; Description of restricted service.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the entity in the target environment if it persists then it updates the same with the input else make an add operation .   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This method represents inquireAllStatusRestrictionsAndExport of the entity @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @return {@link AccountStatusArrayRestrictionResponse}   AccountStatusArrayRestrictionResponse contains array of AccountStatusRestrictionDTO, which will contain following information.  <b>accountStatus</b> account status  <b>accountStatusType</b> account status type  <b>serviceDTOs</b> array of serviceDTO which will contain -   <i>id :</i> ServiceId  <i>name :</i> Name of Service  <i>description :</i> Description of restricted service. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@fcb.param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@fcb.param</b>  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location  This method calls the importBusinessEntity method of the adapter with the requested input once it gets the desired response then calls the exportBusinessEntity method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param bankParametersDTO :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen &lt;br/&gt;This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param bankParametersDTO :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment &lt;br/&gt;This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation . &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param response :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionInquiryResponse} @return :{@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; @param bankParametersDefinitionDTOs :Array of :{@link com.ofss.fc.app.commonservice.core.dto.bankparameter.BankParametersDefinitionDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><b>@param</b>  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.</p> <p><b>@return</b>  List&lt;EntityDTO&gt; entityDTOList</p> <p><b>@throws</b> FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><b>@param</b>  sessionContext :{@link com.ofss.fc.app.context.SessionContext}</p> <p>This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p><b>@param</b>  purposeDTO :{@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO}</p> <p><b>@return</b> :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse}</p> <p><b>@throws</b> :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen &lt;br/&gt;This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param purposeDTO :{@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment &lt;br/&gt;This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation . &lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param response :{@link com.ofss.fc.app.commonservice.core.inquiryresponse.PurposeInquiryResponse} @return :{@link com.ofss.fc.service.response.TransactionStatus} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @param sessionContext :{@link com.ofss.fc.app.context.SessionContext}&lt;br&gt; This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param purposeDTOs :An Array of {@link com.ofss.fc.app.commonservice.core.dto.purposecode.PurposeDTO} @return :{@link com.ofss.fc.service.response.BaseResponse} @throws :{@link com.ofss.fc.infra.exception.FatalException}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the input response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext</code>  <code>,, This parameter captures the basic information about the user of the system.</code>  <code>@fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location. This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputed response else make an add operation .   @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</code></p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p><code>@fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system. &lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse &lt;br&gt; @throws FatalException</code></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.   @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data   @param sessionContext : {@link SessionContext} @param verificationPolicyDTO : {@link VerificationPolicyDTO} @return : {@link VerificationPolicyInquiryResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @param sessionContext : {@link SessionContext} @param response : {@link VerificationPolicyInquiryResponse} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @param sessionContext : {@link SessionContext} @param verificationPolicyDTOs : Array of {@link VerificationPolicyDTO} @return : {@link BaseResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location  This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the   result to file in case of FileDataStore.     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.&lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param List&lt;EntityDTO&gt; entityDTOList @return BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location &lt;br/&gt;This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the &lt;br/&gt; result to file in case of FileDataStore. &lt;br/&gt; &lt;br/&gt; @fcb.param</p> <p>com.ofss.fc.app.context.SessionContext,sessionContext , , This parameter captures the basic information about the user of the system.&lt;br&gt; @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExportList method for the taskCode. This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  List&lt;EntityDTO&gt; entityDTOList @return  BaseResponse @throws FatalException</p>
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		<p>This represents the fetchAndExport method of the entity. This method exports the inquiry response of the entity to the configurable location</p> <p>This method makes an inquiry for the entity for the given key and then calls the adapter for further processing. It calls the audit method for logging into database and export method to export the result to file in case of FileDataStore.</p> <p>@fcb.param  com.ofss.fc.app.context.SessionContext,sessionContext  ,, This parameter captures the basic information about the user of the system.  @fcb.param  domainObjectDto @return entityInquiryResponse  @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAndFetch method of the entity. This method imports the serialised response of the entity from the earlier exported location and returns entity inquiry response which can be used to populate screen  This method calls the importBusinessEntity method of the adapter with the requested input. Once it gets the desired response then calls the audit method of the adapter to make a log of the imported data     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param domainObjectDto @return entityInquiryResponse @throws FatalException
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the importAll method of the entity. This method aims at importing all the exported data for the entity and updating the same in the target environment  This method first fetches the record in the target environment if it persists then it updates the same with the inputted response else make an add operation .     @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param entityInquiryresponse @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IExportImportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ops\IExportImportAdapter	audit		This represents the fetchAndExportList method for the taskCode.This method takes list of DTO as input And fetches the response. For each DTO it calls the fetch method of the application service to get the response. The response is stored in a list. This list is exported to DB/File using export method.    @fcb.param com.ofss.fc.app.context.SessionContext,sessionContext ,, This parameter captures the basic information about the user of the system.  @fcb.param List<EntityDTO> entityDTOList @return BaseResponse @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		Fetches premium amount for a single insuranceDTO @param insuranceDTO @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		@fcb.param sessionContext @fcb.param premiumQuoteRequestDTO @return @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		@fcb.param sessionContext @fcb.param insuranceInstanceId @fcb.param accountId @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		This method is used to fetch the premium quote for the given insurance Id from the premium table - flx_ins_premium_dtls_b @param sessionContext @param premiumQuoteRequestDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		This method updates the QUOTE in PremiumQuoteDetailsDTO[] and POLICY_STATUS in InsuranceDTO.  This is called only for CCI. If the CCI policy is approved , the policy no is generated and set inside InsuranceDTO.  The policy status type inside InsuranceDTO is changed to APPROVED. Also the PremiumStatusType is update to DUE.  In case the CCI policy is not approved, no policy no is generated. Also the policy status type inside InsuranceDTO is set as REJECTED  @param insuranceDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		<p>This method fetches the premium quote for CCI when <a href="#">{@link PremiumPaymentFrequency}</a> is One Time.&lt;br&gt;The following fields are defaulted -&lt;br&gt; &lt;b&gt;1.&lt;/b&gt;&lt;br&gt; <a href="#">{@link PremiumQuoteType}</a> - <a href="#">{@link PremiumQuoteType#QUOTE}</a>&lt;br&gt; &lt;b&gt;2.&lt;/b&gt; <a href="#">{@link PremiumFundingType}</a> - <a href="#">{@link PremiumFundingType#BANK}</a>&lt;br&gt; &lt;b&gt;3.&lt;/b&gt; <a href="#">{@link PremiumPaymentFrequency}</a> - <a href="#">{@link PremiumPaymentFrequency#ONE_TIME}</a>&lt;br&gt; &lt;b&gt;4.&lt;/b&gt; &lt;b&gt;Currency Code - AUD&lt;/b&gt;&lt;br&gt;&lt;br&gt;This methods calculates and set the premium amounts&lt;br&gt; &lt;b&gt;1.&lt;/b&gt; <a href="#">{@link PremiumQuoteDetailsDTO#setBasePremiumAmount(MoneyDTO)}</a> &lt;br&gt; &lt;b&gt;2.&lt;/b&gt; <a href="#">{@link PremiumQuoteDetailsDTO#setNewPremiumAmount(MoneyDTO)}</a> &lt;br&gt; &lt;b&gt;3.&lt;/b&gt; <a href="#">{@link PremiumQuoteDetailsDTO#setInitialPremiumAmount(MoneyDTO)}</a> &lt;br&gt; &lt;b&gt;4.&lt;/b&gt; <a href="#">{@link PremiumQuoteDetailsDTO#setLastPremiumAmount(MoneyDTO)}</a> &lt;br&gt;&lt;br&gt; by invoking <a href="#">{@link #derivePremiumAmount(PremiumQuoteRequestDTO)}</a> which uses the combination of <a href="#">{@link PremiumQuoteRequestDTO#getRequestedLimitAmountDTO()}</a> and <a href="#">{@link CoveredRiskType}</a>&lt;br&gt; If the &lt;b&gt;RequesteLimitAmount&lt;/b&gt; is not a part of the incoming <a href="#">{@link PremiumQuoteRequestDTO}</a> then it is defaulted to <a href="#">{@link #DEFAULT_REQUESTED_LIMIT_AMOUNT}</a></p>
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		<p>@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchPremiumQuote		
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchRefundedPremiumAmount		This method is used to calculate the refunded premium amount @fcb.param sessionContext @fcb.param accountId @return @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchRefundedPremiumAmount		
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchRefundedPremiumAmount		This service is to be consumed by loans. the loans will populate repayment schedule along with other insurance related attributes like borrower details etc as given from origination. @param sessionContext @param premiumQuoteRequestDTO @return
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchRefundedPremiumAmount		Utility method that fetches the refund amount of the cancelled policy, and populates the same in the response object - RefundedPremiumAmountResponse @param sessionContext @param insuranceDTO @return @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	fetchCCIPremiumQuoteForRepaymentSchedule		Sends a quote request to the insurance provider saves the quotes and creates an insurance definition and linkages if any.Returns the response with the aggregated sum of the premiumQuotes. @param insuranceDTO @param cCIConsolidatedPremiumQuoteReqDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	updateQuoteAndPolicyStatus		@fcb.param sessionContext @fcb.param insuranceDTO @fcb.param insuranceLinkageDTOArray @return @throws FatalException
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	cancelLMIPolicy		This method is used to cancelLMIPolicy . The cancellation request can happen in 3 ways  1. Request arrives from origination and facilityId is also passed.   2. Request arrives from origination and only insuranceId is available as part of incoming insurance entity object. FacilityId is not passed in this case.   3. Request arrives from the LCM09 screen.   The call made for collection happens in only the 3rd case. Also for 3rd case , all the policies except those in PolicyStatusType.PLC_CANCEL is to be allowed.   So an extra check is put  <t><b>(isStandAloneCall    (!isStandAloneCall && insurance.getPolicyStatusType() != PolicyStatusType.PENDING_CANCEL))</b></t>   to distinguish 1st n 2nd case from the 3rd case.
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	cancelLMIPolicy		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	cancelLMIPolicy		This method is used to cancelLMIPolicy . The cancellation request can happen in 3 ways  1. Request arrives from origination and facilityId is also passed.   2. Request arrives from origination and only insuranceId is available as part of incoming insurance entity object. FacilityId is not passed in this case.   3. Request arrives from the LCM09 screen.   The call made for collection happens in only the 3rd case. Also for 3rd case , all the policies except those in PolicyStatusType.PLC_CANCEL is to be allowed.   So an extra check is put  <t><b>(isStandAloneCall    (!isStandAloneCall && insurance.getPolicyStatusType() != PolicyStatusType.PENDING_CANCEL))</b></t>   to distinguish 1st n 2nd case from the 3rd case.
IInsInsuranceProviderAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsInsuranceProviderAdapter	cancelLMIPolicy		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		Returns DocumentInquiryResponse for documentID
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		Return DocumentInquiryResponse for a Index type and Index Value pair

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		Returns 'true' if DocumentId is valid, else returns 'false' @param documentId @return DocumentID Validity For Collateral Or Party.
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		Fetches the List of Invalid Document Id's @param partyId @param collateralId @param documentIds @return invalidDocumentIds
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTO		Returns the fetched Document ID with content with bytes @param documentId @return documentDTO
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchDocumentDTOForIndex		Return DocumentInquiryResponse for a Index type and Index Value pair

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	retrieveRendition		
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	retrieveRendition		Returns the fetched Document ID with content with bytes @param documentId @return documentDTO
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	addDocument		This method is used to <u>add</u> response from third party interface or if mode is manual key in of report from collateral valuation report screen. @param SessionContext aSessionContext : holds the basic information about the user of the system. @param PropertyValuationResponseDTO aResponseDTO : Property Valuation response for the transaction. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.report.response.CollateralValuationResponseInquiryResponse} IResponse : This class holds all the values of dtos which are defined in it.   @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	addDocument		This method is used to update a valuation report which has already been stored.  This method is used in the case when amendment of a valuation report is performed.   @param SessionContext aSessionContext : This parameter captures the basic information about the user of the system.   The information is used by the system for authentication purposes. @param PropertyValuationResponseDTO aResponseDTO : This DTO contains the details regarding the valuation report for which amendment has been performed. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.report.response.CollateralValuationResponseInquiryResponse} This class holds all the values of dtos which are defined in it.   @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForNewAndAmend		Generate Covering Letter for Collateral Valuation. The OutboundDocumentRequest DTO is varying according the request document type. It contains either CollateralValuationRequest or CollateralTitleSearchRequest auxiliary index type. And auxiliary index value is also varying according to request document type. It also contains the parameters which have to be passed to BIP to generate the covering letter. All parameters are mandatory. OutboundDocumentRequestDTO need to be set the below parameters:- 1.reportRequestDTO.setDocumentType(Request constants from com.ofss.fc.common.lcm.LCMCoveringLetterConstants). 2.auxiliaryDocumentIndexKeyDTO.setIndexType(Auxiliary Index Type constants from com.ofss.fc.common.lcm.LCMCoveringLetterConstants); auxiliaryDocumentIndexKeyDTO.setIndexValue(It is coming from service according document type);

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>auxiliaryDocumentIndexDTO.setKeyDTO(auxiliaryDocumentIndexKeyDTO);  auxiliaryDocumentIndexDTO.setIsPrimaryIndex(isPrimaryIndex constants from  com.ofss.fc.common.lcm.LCMCoveringLetterConstants);  3.documentDTO.setAuxiliaryDocumentIndexes(auxiliaryDocumentIndexDTO);  4.outboundDocumentRequestDTO.setReportRequestDTO(reportRequestDTO);  5.outboundDocumentRequestDTO.setDocumentDTO(documentDTO);This method is called to  generateCoveringLetterForCollateralValuationNewAndAmend the request DTO for generating PDF  @param outboundDocumentRequestDTO the document input dto @return the  com.ofss.fc.app.content.outbound.dto.DocumentGenerationResponse @throws FatalException the fatal  exception</p>
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForNewAndAmend		<p>This method is used to &lt;u&gt;add&lt;/u&gt; the valuation request to the valuation draft table. &lt;br&gt; The collateral valuation DTO is stored in the table in the form of a blob with the key as the valuation reference number. @param SessionContext aSessionContext : holds the basic information about the user of the system. @param CollateralValuationDTO collateralValuationDTO : Collateral valuation request details DTO. @param String draftStatus : The status of the current valuation request(INITIATED/UNINITIATED). Will be uninitiated before adding the request to the main request table. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.CollateralValuationInquiryResponse} response : This class holds all the values of dtos which are defined in it.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				  @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForNewAndAmend		
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForQueryRequest		This method <b>returns</b> the details of the <b>Queries</b> initiated on a particular <b>Report Reference</b>. @param SessionContext aSessionContext : holds the basic information about the user of the system. @param CollateralValuationQueryDTO collateralValuationQueryDTO : Valuation query details for the valuation report. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.report.query.CollateralValuationQueryInquiryResponse} This class holds all the values of dtos of queries which are defined in it.   @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForQueryRequest		This method is called to generate Covering Letter For Collateral Valuation Query Request the request DTO for generating PDF @param collateralValuationQueryDTO @return documentGenerationResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForTitleSearchCancelRequest		This method <u>Cancels</u> the TitleSearch request for the given request message reference . @fcb.param SessionContext,sessionContext, Holds the basic information about the user of the system. @fcb.param CollateralTitleSearchCancellationDTO,cancellationDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForTitleSearchNewAndAmend		
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForTitleSearchNewAndAmend		
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForTitleSearchNewAndAmend		This method is called to generate CoveringLetter For TitleSearch New And Amend Request the request DTO for generating PDF @param collateralTitleSearchDTO @return documentGenerationResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchAllDocuments		This method is used to <u>add</u> response from third party interface or if mode is manual key in of report from collateral valuation report screen. @param SessionContext aSessionContext : holds the basic information about the user of the system. @param PropertyValuationResponseDTO aResponseDTO : Property Valuation response for the transaction. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.report.response.CollateralValuationResponseInquiryResponse} IResponse : This class holds all the values of dtos which are defined in it.   @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchAllDocuments		This method is used to update a valuation report which has already been stored.  This method is used in the case when amendment of a valuation report is performed.   @param SessionContext aSessionContext : This parameter captures the basic information about the user of the system.   The information is used by the system for authentication purposes. @param PropertyValuationResponseDTO aResponseDTO : This DTO contains the details regarding the valuation report for which amendment has been performed. @return {@link com.ofss.fc.app.lcm.dto.collaterals.valuation.report.response.CollateralValuationResponseInquiryResponse} IResponse : This class holds all the values of dtos which are defined in it.   @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	fetchAllDocuments		fetches all documents. @return IRes @see com.ofss.fc.app.adapter.lcm.collaterals.valuation.ICollateralValuationContentAdapter#fetchAllDocuments(java.lang.String, java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForCancelRequest		This method is called to generateCoveringLetterForCollateralValuationCancelRequest the request DTO for generating PDF @param collateralValuationCancelConfirmationDTO @return documentGenerationResponse @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForCancelRequest		This method is called to generate CoveringLetter For Title Search Cancel Request the request DTO for generating PDF @param collateralValuationCancellationDTO @return documentGenerationResponse @throws FatalException
ICollateralValuationContentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\valuation\ICollateralValuationContentAdapter	generateCoveringLetterForCancelRequest		This method <u>Cancels</u> the valuation request for the given request message reference . @param SessionContext sessionContext : holds the basic information about the user of the system. @param CollateralValuationCancellationDTO cancellationDTO : Collateral valuation request cancellation details DTO. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	<p>This method will load the party bundle linkage with the input bundle reference id and perform delink operationValidation for the input list of account ids. </p> <code>@fcb.param MI, AccountDetailsDTO[], accountDetailsDTOArray, Indicates the list of accounts on which the action is to be performed. @param fetchedPartyBundleLinkage @param allBundleAccountLinkageList @param delinkBundleAccountLinkageArray @fcb.param String nominatedAccountId, Indicates the nominated account for the bundle. In amend flow, if the nominated account has changed, this field should contain the new nominated account, else the earlier value.</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	<p>This method will load the party bundle linkage with the input bundle reference id and perform delink operationValidation for the input list of account ids. </p> <code>@fcb.param MI, AccountDetailsDTO[], accountDetailsDTOArray, Indicates the list of accounts on which the action is to be performed. @param fetchedPartyBundleLinkage @param allBundleAccountLinkageList @param delinkBundleAccountLinkageArray @fcb.param String nominatedAccountId, Indicates the nominated account for the bundle. In amend flow, if the nominated account has changed, this field should contain the new nominated account, else the earlier value.</code>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> <code>@return {@link IBundleAccountAdapter}</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> <code>@return {@link IBundleAccountAdapter}</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> <code>@return {@link IBundleAccountAdapter}</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>&lt;p&gt;This method will fetch details of the input account id. &lt;/p&gt;  @fcb.param MI, accountId @return {@link AccountInquiryResponse}  @throws FatalException</p>	<p>&lt;p&gt;This method will populate un-associated account details for CASA accounts. &lt;/p&gt;  @fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>&lt;p&gt;This method will fetch details of the input account id. &lt;/p&gt;  @fcb.param MI, accountId @return {@link AccountInquiryResponse}  @throws FatalException</p>	<p>&lt;p&gt;This method will populate un-associated account details for CASA accounts. &lt;/p&gt;  @fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>This method will populate un-associated account details for Credit Card accounts. </p> @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>This method will populate un-associated account details for Credit Card accounts. </p> @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>&lt;p&gt;This method will fetch details of the input account id. &lt;/p&gt;  @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</p>	<p>&lt;p&gt;This method will populate unassociated account details for LOAN accounts. &lt;/p&gt; @fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <p>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</p>	<p>This method will populate unassociated account details for LOAN accounts. </p> <p>@fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <p>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</p>	<p>This method will populate unassociated account details for TD accounts. </p> <p>@fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	<p>This method will populate unassociated account details for TD accounts. </p> @fcb.param MI, SessionContext, sessionContext, Standard header information for the application. @fcb.param MI, String, bundleId, Indicates the bundle id chosen for subscription. @fcb.param MI, SubscriptionDetailsDTO[], subscribedOfferDetailsDTO, Indicates the offer details within the chosen bundle and also the applicants for each offer. @fcb.param MI, String[], partyIdArray, Indicates the list of party id(s) for which all accounts are to be fetched. @fcb.param MI, PartyToAccountRelationshipDTO, partyAccountRelationshipDTO, Indicates the party account relationship fetched after inquiring details for input party id(s) from party module. @return {@link com.ofss.fc.app.account.offerbundling.dto.inquiry.BundleOwnerAccountDetailsDTO} @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	This method will fetch all unassociated accounts of the identified bundle owners. It will include accounts which satisfy the following  <ul> <li>not linked to any other bundle</li> <li>linked to a bundle but the bundle status is Expired/Unsubscribed</li> <li>status other than closed, marked for closure, written off and matured</li> </ul> @fcb.param MI, bundleOwner @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	This method will fetch all unassociated accounts of the identified bundle owners. It will include accounts which satisfy the following <ul> <li>not linked to any other bundle</li> <li>linked to a bundle but the bundle status is Expired/Unsubscribed</li> <li>status other than closed, marked for closure, written off and matured</li> </ul> <code>@fcb.param MI, bundleOwner @throws FatalException</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	For the subscribed account, this method will populate account specific data.  <code>@fcb.param MI, BundleOwnerAccountDetailsDTO[], bundleOwnerAccountDetailsDTO, Indicates the account details held by the bundle owner. @fcb.param MI, int, index, Indicates the index at which to set values. @fcb.param MI, BundleAccountLinkage, bundleAccountLinkage, Indicates the bundle account linkage which belongs to the subscription. @return @throws FatalException</code>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> <code>@fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException</code>	For the subscribed account, this method will populate account specific data.  <code>@fcb.param MI, BundleOwnerAccountDetailsDTO[], bundleOwnerAccountDetailsDTO, Indicates the account details held by the bundle owner. @fcb.param MI, int, index, Indicates the index at which to set values. @fcb.param MI, BundleAccountLinkage, bundleAccountLinkage, Indicates the bundle account linkage which belongs to the subscription. @return @throws FatalException</code>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	This method will fetch the details of each supplied account id.  @fcb.param MI, AccountDetailsDTO[], accountDetailsDTOArray, Indicates the list of account ids to be linked to the bundle at the time of bundle subscription. @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	fetchAccountDetails	<p>This method will fetch details of the input account id. </p> @fcb.param MI, accountId @return {@link AccountInquiryResponse} @throws FatalException	This method will fetch the details of each supplied account id.  @fcb.param MI, AccountDetailsDTO[], accountDetailsDTOArray, Indicates the list of account ids to be linked to the bundle at the time of bundle subscription. @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	This method will check if the input account number is in fixed rate of interest or variable. @param accountId @param accountModule @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	This method will check if the input account number is in fixed rate of interest or variable. @param accountId @param accountModule @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance.</p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.</p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.</p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInFixedRate	This method will load the demand deposit account and check if the account being added to the bundle is in fixed rate period of variable. @fcb.param MI, accountId @return @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	From release 2.3 Bundle and Facility fee has a different behaviour during hardship relief. The Bundle/Facility fee will be deferred for if the relief type is "WaiveFee". In case of non Bundle/Facility fee it will follow the normal waive fee behaviour.
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if the input account is in hardship. @fcb.param MI, String, accountId, Indicates the account id. @fcb.param MI, TaskModuleType, accountModule, Indicates the module to which the input account id belongs. @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	This method will check if the input account is in hardship. @fcb.param MI, String, accountId, Indicates the account id. @fcb.param MI, TaskModuleType, accountModule, Indicates the module to which the input account id belongs. @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isAccountInHardship	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	This method validates that the nominated account selected for the bundle must be a CASA or LOAN account. @param bundleAccountLinkage @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	This method validates that the nominated account selected for the bundle must be a CASA or LOAN account. @param bundleAccountLinkage @return
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	isNominationAllowed	@fcb.param MI, accountId, String, Indicates if the input account can be nominated for bundle fee charging. @return @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLink Account	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLink Account	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLink Account	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyLinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyDelinkAccount	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @param isAccountDelinked @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.</p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.</p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleExpiry	TODO @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyBundleUnsubscription	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>This method is used by account modules (CASA/TD/LOAN) to intimate bundle servicing of any rate change action happening on the account which belongs to a bundle and has bundle benefit applicability set to true.&lt;br /&gt; &lt;br /&gt; The boolean isFixedRate indicates the interest rate in which the account going.&lt;br /&gt; &lt;br /&gt; This is required because transition of account from fixed to variable or variable to fixed rate may affect the bundle benefits applicable to the account. The bundle servicing module will derive the bundle applicability status after this event and if there is a change, it will intimate account module to update the same in pricing.&lt;br /&gt; @Since R2.2 @fcb.param MI, String, accountId, Indicates the identifier for the account being acted upon. @fcb.param MI, {@link TaskModuleType}, accountModule, Indicates the module to which the account belongs. @fcb.param String, offerId, Indicates the offer under which the account has been opened. @fcb.param String, bundleId, Indicates the unique identifier for bundle to which the account belongs. @fcb.param MI, String, bundleReferenceId, Indicates the bundle reference id to which the account belongs. @fcb.param MI, boolean, isFixed, Indicates if the account is in fixed rate or variable. @fcb.param MI,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				{@link Date}, asOnDate, Indicates the date on which the rate change action is taking place.
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO</p> <p>@param accountId  @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>This method is used by account modules (CASA/TD/LOAN) to intimate bundle servicing of any rate change action happening on the account which belongs to a bundle and has bundle benefit applicability set to true.&lt;br /&gt; &lt;br /&gt; The boolean isFixedRate indicates the interest rate in which the account going.&lt;br /&gt; &lt;br /&gt; This is required because transition of account from fixed to variable or variable to fixed rate may affect the bundle benefits applicable to the account. The bundle servicing module will derive the bundle applicability status after this event and if there is a change, it will intimate account module to update the same in pricing.&lt;br /&gt; @Since R2.2 @fcb.param MI, String, accountId, Indicates the identifier for the account being acted upon. @fcb.param MI, {@link TaskModuleType}, accountModule, Indicates the module to which the account belongs. @fcb.param String, offerId, Indicates the offer under which the account has been opened. @fcb.param String, bundleId, Indicates the unique identifier for bundle to which the account belongs. @fcb.param MI, String, bundleReferenceId, Indicates the bundle reference id to which the account belongs. @fcb.param MI, boolean, isFixed, Indicates if the account is in fixed rate or variable. @fcb.param MI, {@link Date}, asOnDate, Indicates the date on which the rate change action is taking place.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. &lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	<p>TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException</p>	<p>&lt;p&gt;If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance.&lt;/p&gt; @return {@link IBundleAccountAdapter}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyRateTypeChange	TODO @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>This method does the processing when a request to change nominated account for an existing bundle reference id is made. </p> @param partyBundleLinkage @param componentPreferenceDTO
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>This method does the processing when a request to change nominated account for an existing bundle reference id is made. </p> @param partyBundleLinkage @param componentPreferenceDTO
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p> If the new account being nominated is a CASA or LOAN account, it is required to only update the data, where as if the new nominated account is a LINKED account, a new entry is to be created for bundle account linkage. </p> @param partyBundleLinkage @param bundleAccountLinkage @param accountModule @param accountBasicDetailsDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p> If the new account being nominated is a CASA or LOAN account, it is required to only update the data, where as if the new nominated account is a LINKED account, a new entry is to be created for bundle account linkage. </p> @param partyBundleLinkage @param bundleAccountLinkage @param accountModule @param accountBasicDetailsDTO
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is CASA, this method will check if the Bundle-DDA adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is LOAN, this method will check if the Bundle-LOAN adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is TD, this method will check if the Bundle-TD adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	<p>If the input task module type is Cards, this method will check if the Bundle-Card adapter is mocked and based on the mocking constant, return new instance. </p> @return {@link IBundleAccountAdapter}
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity
IBundleAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\bundle\IBundleAccountAdapter	notifyNominationChange	@fcb.param MI, accountId, String, Indicates the account which needs to be checked if is in hardship. @fcb.param MI, isCurrentlyNominated, boolean , Indicates if the account is currently nominated for bundle fees. @throws FatalException	@param partyBundleLinkage @param bundleAccountLinkageList @param actionProcessingDate @param bundleActivity

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductRoleHeadAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IProductRoleHeadAdapter	isAccountHeadValid		
IProductRoleHeadAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IProductRoleHeadAdapter	getLedgerDetails		
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForTeller	This method fetches count of unapproved work items that match the date criteria specified by the user   depending on the date parameter. @param valueDate @return @throws FatalException	This method will check if any transactions are pending for the teller. The method will exclude any future dated currency indents.
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForTeller	This method fetches count of unapproved work items that match the date criteria specified by the user   depending on the date parameter. @param valueDate @return @throws FatalException	This method will validate workitem status. @param batchOperationDTO @param sessionContext
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForTeller	This method fetches count of unapproved work items that match the date criteria specified by the user   depending on the date parameter. @param valueDate @return @throws FatalException	
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForRole	This method fetches count of unapproved work items that match the currency criteria specified by the user   depending on the transaction currency parameter. @param currencyCode @return @throws FatalException	This method will validate if any transactions are pending for the currency which is getting deleted/ineffective.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForRole	This method fetches count of unapproved work items that match the currency criteria specified by the user   depending on the transaction currency parameter. @param currencyCode @return @throws FatalException	
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForUser	@param postingDate @param tellerId @return @throws FatalException	This method will check if any transactions are pending for the teller. The method will exclude any future dated currency indents.
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForUser	@param postingDate @param tellerId @return @throws FatalException	This method will validate workitem status. @param batchOperationDTO @param sessionContext
IBranchWorklistAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchWorklistAdapter	fetchUnapprovedTransactionForBranch		
IPartyKYCStatusAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyKYCStatusAdapter	PartyKYCStatusConfirmed		
IPartyKYCStatusAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyKYCStatusAdapter	PartyKYCStatusConfirmed		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchProductDetails	This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired @return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws FatalException	
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchProductDetails	This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired @return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws FatalException	This represents the changePartyToAccountRelationship method of the entity. This method handles the following scenarios.  1. If there is no {@code PartyToAccountRelationshipDTO} existing for the account i.e new account  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;=> If the input is one or more new {@code PartyToAccountRelationshipDTO} then it will add the {@code PartyToAccountRelationshipDTO}.  2. If there are {@code PartyToAccountRelationshipDTO} already exist for an account. Lets consider there are 5 Parties (A, B, C, D & E) linked to the account CASA0001.  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;=> CASE 1: If the input has all 5 {@code PartyToAccountRelationshipDTO} then it will update all 5 {@code PartyToAccountRelationshipDTO}.  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;=> CASE 2: If the input has all 5 {@code PartyToAccountRelationshipDTO} and 2 of the



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchLoanO fferDTO		This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray
IPartyPMUA dap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This represents the checkMandatoryRelationshipForPartyType method @fcb.param MI.com.ofss.fc.app.context.SessionContext,sessionCon text, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,RelationshipRuleDTO,relationshipRuleDTO, ,relationshipRuleDTO TODO: Explain the relationshipRuleDTO above @return {@link DomainObjectDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method is used to fetch the current status of a party whether it is a minor, operational major or legal major depending on the date of birth and the current posting date. @param dateOfBirth @param postingDate @return
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	Method to calculate the KYC Expiry Date for a given record. It uses the expiry duration defined as part of BankPolicyDefinition. @param bankCode @param postingDate @return Date kycExpiryDate
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method is used to log the kyc expiry action in the party actions table as and when the kyc status is set as 'Confirmed' and an account under the particular partyId is marked as 'Regular'. It will make a call to the {@link PartyActionLoggerService#logKYCUpgradationAction (String, Date, String, long, String)} to log the actions. The branchCode for the partyId is obtained from the {@link BrandDemograph} or from the {@link FCRThreadAttribute} @param partyId @param expiryDate @param serialNumber @param determinantValue
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchPartyB ankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method will set the PartyAge status string and the party "major" flags. @param birthDate @param partyType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method is used to calculate the status change date and log an action for status change on that day. During party on boarding while adding the legislative demographics, if the party status is minor, then this method calculates the date when the minor turns to operational major or legal major and logs an action in the party actions table which will be processed during the batch processing and the legislative demographic is updated. @param party @param status @return
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchDDAOfferDetails	@param offerCode @return	
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchDDAOfferDetails	@param offerCode @return	This method fetches maximum number of accounts allowed for the given offer for a single party. @param sessionContext @param accountType @param productCode @return @throws FatalException
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchTDOfferChannelFacilities		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchTDOff erChannelF acilities		This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchDDOff erChannelF acilities		
IPartyPMUAdap ter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \party\IPartyPMUAdap ter	fetchDDOff erChannelF acilities		This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchOfferDetails		
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchOfferDetails		
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchOfferDetails		
IPartyPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyPMUAdapter	fetchOfferDetails		This service validates the incoming party to account relationship channel facility flags against the account. Further if any affiliates are linked to the account, the same channel flags are further compared to the allowed access channels on relationship.     For cases where flag areToBeDefaulted is set to true, no validation is to be performed. The default values of the flags are stamped against the party account relationship.     For TRU and LEG party types, areToBeDefaulted is default set to true. No validation should be performed as channel facilities are not applicable for these.    New /Delink existing parties from stand alone party to account relationship screen is not allowed for Lending Accounts i.e. Loan or OD. @param partyToAccountRelationshipArray



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDCertificatePrintAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDCertificatePrintAdapter	printDepositCertificate		This method will print term deposit certificates and update Deposit Certificate Status in Deposit table. System prints the selected certificate(s) having the following data elements. <ul> <li>Account Holder's Name</li> <li>TD Account number</li> <li>Account branch</li> <li>Deposit number</li> <li>Net Interest Rate</li> <li>Principal Balance</li> <li>Maturity Date</li> <li>Maturity Amount</li> </ul> @param sessionContext @param depositCertificateDTO @return TransactionStatus @throws FatalException
ITDCertificatePrintAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDCertificatePrintAdapter	printDepositCertificate		This method calls the OutboundDocumentApplicationService.generateSetOfDocument for printing. @param sessionContext @param certificatePrintFactsDTO
ISimulationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationAccountAdapter	formActionSequence	This method is used to form the new action sequence using the modified actions set @param applicationContext @param account @return	This method is used to skip the actions for the given module type @param actionsToBeSkipped @param abstractBatchData @param taskModuleType
ISimulationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\simulation\ISimulationAccountAdapter	formActionSequence	This method is used to form the new action sequence using the modified actions set @param applicationContext @param account @return	This method is used to form the action sequence from the actions
IAccountTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDAdapter	isAccountValid	@param accountId @throws FatalException	This method will validate the account id on the basis of the account Id. @param accountId @param accountType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isAccountValid	@param accountId @throws FatalException	<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object {@link TitleModificationResponseDTO}. Add the array of account ids by calling toArray() on the arraylist. &lt;li&gt;Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException</li> </ul> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isAccountValid	@param accountId @throws FatalException	This method checks the validity of the specified account number.   It is invoked by every maintenance method of this service to check whether the specified account is valid or whether it exists or not.  @param sessionContext @param accountId @param accountType @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isAccountValid	@param accountId @throws FatalException	This method fetches all the deposits for which the deposits have been maintained and do not have their status as closed and their respective deposit details. Following are the steps to be followed in this method:-  <ol> <li>Create an instance of AccountNominationsResponse </li>Fetch the transaction status using fetchTransactionStatus().</li> <li>Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments</li> <li>Check for the existence of the account by calling fetchAccount() of TermDepositAccountApplicationService through the TD adapter. </li>Create an instance of Domain Service and invoke it's fetchAllNominatedDeposits() method by passing accountId. This would return an array of depositNumbers.</li> <li>For each deposit in the array fetched,call fetchDeposit() of DepositApplicationService through the TD adapter. Set each response in the Inquiry Response if the deposit status is not closed. </li>The inquiry response will not have any party details. </li>Fill the Transaction Status by calling the method fillTransactionStatus and pass the transaction status as an argument to it.</li> <li>Set the response status as transaction status</li> <li>Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks.</li> <li>Close the interaction in finally block.</li> <li>return the Response of this operation.</li> </ol> / @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				sessionContext @param accountId @param accountType @return AccountNominationsResponse @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchDepositDetails	@param accountId @param depositNumber @return @throws FatalException	This method handles {@link TDActionType#Statement} actions that have been logged. It is responsible for generating the statements that are part of the statement request period. The method follows the below steps: <ol> <li>Fetch the account details and the statement parameter details using the {@link IAccountDDAAdapter} and {@link IAccountProductAdapter} respectively. <li>Get all the statement requests that need to be generated using {@link StatementRequestService#fetchTDStatementRequestOnActionDueDate(Date, String)} method. <li>Loop through each of the statement request and do the following: <ol> <li>If the current statement request is reorder, then update the statement request with StatementGenerationStatus = ReOrdered and corresponding remarks. StatementStatisticsDTO must be null. Continue to the next request. <li>If StatementRequestType = SystemInitiatedRequest then: <ol> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus. <li>If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details. <li>Call the {@link IAccountTDA dapter#modifyStatementAttributes()} to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>update statement details at account level. &lt;li&gt;The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request. &lt;/ol&gt; &lt;li&gt;For customer initiated statements,generate the statement using {<a href="#">@link #generateStatementRequest()</a>} and update the statement request generation details. Continue to the next request. &lt;/ol&gt; &lt;li&gt;The same process will be repeated for all the remaining statement requests. &lt;/ol&gt; @param applicationContext @param actionKey @param actionRequest @param dueDate</p>
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchDepositDetails	@param accountId @param depositNumber @return @throws FatalException	<p>&lt;p&gt;This method is responsible for processing Statement Generation during End of Day activity.&lt;br&gt;&lt;/p&gt; @param applicationContext @param accountId @param depositId @param statementRequestID</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchDepositDetails	@param accountId @param depositNumber @return @throws FatalException	<p>This method used to create term deposit statement request. Customer or system can initiate statement request. statement request can be for demand deposit or term deposit account. System can initiate request using statement setup done at product/offering level or overridden a account level. Customer can ask for new statement or reorder existing statement . Statement request generally contain</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Period - Start and End date</li> <li>&lt;li&gt;Delivery mode details</li> <li>&lt;li&gt;Service charge details.</li> </ul> <p>&lt;p&gt;The method follows the following algorithm:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Set the statement request key from {@link UUIDGenerator#generateUuid()} &lt;li&gt;Set the statementRequestStatus as pending.</li> <li>&lt;li&gt;Create a new instance of {@link AbstractBusinessPolicy} based on the instance type of statementRequest and validate the same.</li> <li>&lt;li&gt;The statement requests will be placed at the deposit level as it an account level statement request may cause inconsistencies during batch. Hence for the current account, all the deposits are fetched and statement request is placed for each with {@link TDStatementRequest#accountBasedStmtId} as the same.</li> <li>&lt;li&gt;For System initiated request, make sure that the {@link StatementRequest#getCustomerInitiatedRequest()} is initialized with all the components as null so that isReorder flag is not persisted as null.</li> <li>&lt;li&gt;Call the {@link #create(AbstractDomainObject, com.ofss.fc.framework.repository.AbstractRepository)} method with instance of {@link StatementRequestRepository}</li> </ul> <p>&lt;/ol&gt; @param tdStatementRequest @param partyId, Denotes the partyId @param branchId, Denotes the branchId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchDepositDetails	@param accountId @param depositNumber @return @throws FatalException	This method fetches the deposit details for all the deposits present in the input entity array. @param sessionContext @param accountId @param nominations @return @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchParentDepositNumber	@param accountId @param depositNumber @return @throws FatalException	
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	validateAccountAndFetchProductCode	@param accountId @param depositNumber @return @throws FatalException	This method fetches the product code for the input account Id. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method will return the primary account holder of the account, @param sessionContext @param accountId @param accountType @return String
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	This method should be used by all internal modules for inquiring the account. This will avoid Interaction begin and close related calls. @param sessionContext @param accountId @param accountType @return @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAccountDetails	@param sessionContext @param accountId @return @throws FatalException	The following method is used to fetch account related information.The following method is used to avoid Interaction begin and close related operations. @param accountId @param accountType @return @throws FCRException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	getAccountInformation	@param sessionContext @param accountId @return @throws FatalException	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	modifyTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	modifyTitle	@param sessionContext @param accountId @param modifiedTitle @return @throws FatalException	<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; &lt;li&gt;Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object {@link TitleModificationResponseDTO}. Add the array of account ids by calling toArray() on the arraylist. &lt;li&gt;Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDAadapter	modifyStatementAttributes	@param accountId @param period @throws FatalException	<p>This method processes system initiated statement request. It does the following:</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>&lt;li&gt;If it is found to be valid, then generate the statements using {@link #generateStatementRequest()} method. Call the pricing module to charge for the service and update statement request generation details.</li> <li>&lt;li&gt;Call the {@link IAccountTDAadapter#modifyStatementAttributes()} to update statement details at account level.</li> <li>&lt;li&gt;The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request.</li> </ul> </li> </ul> <p>&lt;/ol&gt; @param currentStatementRequest @param depositDetails @param statementParametersDTO @param adapterDepositDetailsDTOs2</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	modifyStatementAttributes	@param accountId @param period @throws FatalException	<p>This method processes system initiated statement request. It does the following:</p> <ul style="list-style-type: none"> <li>Validate the account's user and system defined status and check if they are not blocked in statement parameters. If they are found to be blocked, then update the statement request with StatementGenerationStatus = NotExecutedDueToInvalidStatus.</li> <li>If it is found to be valid, then generate the statements using <a href="#">{@link #generateStatementRequest()}</a> method. Call the pricing module to charge for the service and update statement request generation details.</li> <li>Call the <a href="#">{@link IAccountTDAadapter#modifyStatementAttributes()}</a> to update statement details at account level.</li> </ul> <p>The next system initiated statement request needs to be placed based on the statement frequency in statement parameters. Check that the account status is not closed or inactive and then place the next order. continue to next request.</p> <p>@param currentStatementRequest @param depositDetails @param statementparametersDTO @param adapterDepositDetailsDTOs2</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isExistingAccount	@param sessionContext @param accountId @throws FatalException	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest} .Customer can reorder existing generated/requested statement.This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link DDStatementRequestDTO},ddStatementRequestDTO , It contains all the information required to place a</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending</li> <li>statementNo : Must be set to null for statement order.It will be set at statement generation.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null</li> </ul> <p>@return {@link DDStatementRequestInquiryResponse}. It contains the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link DDStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li>{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</li> </ol> <p>@throws FatalException</p>
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isExistingAccount	@param sessionContext @param accountId @throws FatalException	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest} .Customer can reorder existing generated/requested statement.This method is used to order term deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,{@link TDStatementRequestDTO},tdStatementRequestDTO , , It contains all the information required to place a demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>next working day.</p> <ul style="list-style-type: none"> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.</li> <li>statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.</li> <li>statementNo : Must be set to null for statement order.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.</li> </ul> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDueDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</p> <ul style="list-style-type: none"> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus :{@link StatementRequestStatus} must be null. Will be set to pending.</li> <li>statementNo : Must be set to null for statement order.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null</li> </ul> <p>@return {@link TDStatementRequestInquiryResponse}. It contains the {@link TDStatmentRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li><b>Transaction Success</b> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> </li> <li><b>Transaction Failure</b> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> </li> <li><b>Error Reason</b> may be one of the following <ul style="list-style-type: none"> <li>{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</li> </ul> </li> </ul> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountTDA dapter	isExistingAccount	@param sessionContext @param accountId @throws FatalException	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest} .Customer can reorder existing generated/requested statement.This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link LoanStatementRequestDTO},loanStatementRequestDT O,, It contains all the information required to place a</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key : contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : StatementPeriodType MI. Denotes the period type of the input.</li> <li>period : Period MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : Date MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : Date must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing : FeeDetailsDTO contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType : StatementRequestType, MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus : StatementRequestStatus must be null. It will be set to pending.</li> <li>statementNo : Must be set to null for statement order. It will be set at statement generation.</li> <li>customerInitiatedRequest : CustomerInitiatedRequest Must be null.</li> </ul> <p>@return LoanStatementRequestInquiryResponse. It contains</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the {@link LoanStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li>{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</li> </ol> <p>@throws FatalException</p>
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAllDepositDetails	@param accountId @return @throws FatalException	<p>This method will validate if pay-in has been done for the input term deposit account.</p> <p>@fcb.param MI, BundleSubscriptionAction, BundleSubscriptionAction, Indicates the action being performed. @fcb.param MI, String, accountId, Indicates the account id for which pay-in is to be validated. @fcb.param MI, TaskModuleType, accountModule, Indicates the module to which the input account id belongs. @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAllDepositDetails	@param accountId @return @throws FatalException	<p>Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. &lt;/br&gt;Consider the following example. &lt;/br&gt;AccountId: TestAccount001 &lt;/br&gt;Opening date: 22nd-feb-2008 &lt;/br&gt;Statements generated: &lt;table border='1'&gt; &lt;tr&gt; &lt;th&gt;Statement no.&lt;/th&gt; &lt;th&gt;Statement period start&lt;/th&gt; &lt;th&gt;Statement period end&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;nth&lt;/td&gt; &lt;td&gt;1-april-2008&lt;/td&gt; &lt;td&gt;30-april-2008&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1&lt;/td&gt; &lt;td&gt;22-feb-2008&lt;/td&gt; &lt;td&gt;29-feb-2008&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</p> <p>Suppose the current process date is 24th may 2008.&lt;/br&gt;&lt;/br&gt;Then the current period: 1-may-2008 to 24th-may-2008 &lt;/br&gt;Prior statement period is: 1-april-2008 to 30-april-2008 &lt;/br&gt;&lt;/br&gt;In case there are no statements generated till 24th may 2008, then &lt;/br&gt;Current statement period: 22nd-feb-2008(opening date) to 24th may 2008 &lt;/br&gt;Prior statement period:22nd-feb-2008(opening date) to 24th may 2008&lt;p&gt;This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as follows: &lt;ol&gt; &lt;li&gt;start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>be after the current process date. &lt;li&gt;end date: Its the current process date. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId , , It is the account id of the account for which statement period is to be fetched. @fcb.param MI,{@link com.ofss.fc.enumeration.account.AccountType},accountType, , It denotes the type of account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit. @return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchAllDepositDetails	@param accountId @return @throws FatalException	This method will fetch all the deposit details for the given accountId. @param accountId @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteDeposit Details	@param accountId @return	This method will return all the details of a deposit that need to be displayed in the statement inquiry screen. @param sessionContext @param accountId @param depositNumber @param interestClass @param interestDivision @param effectiveDate @throws FatalException
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteDeposit Details	@param accountId @return	Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated.  Consider the following example.  AccountId: TestAccount001  Opening date: 22nd-feb-2008  Statements generated: <table border='1'> <tr> <th>Statement no.</th> <th>Statement period start</th> <th>Statement period end</th> </tr> <tr> <td>nth</td> <td>1-april-2008</td> <td>30-april-2008</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1</td> <td>22-feb-2008</td> <td>29-feb-2008</td> </tr> </table> Suppose the current process date is 24th may 2008.  Then the current period: 1-may-2008 to 24th-may-2008  Prior statement period is: 1-april-2008 to 30-april-2008   In case there are no statements generated till 24th may 2008, then  Current statement period: 22nd-feb-2008(opening date) to 24th may 2008  Prior statement period:22nd-feb-2008(opening date) to 24th may 2008<p>This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>follows: &lt;ol&gt; &lt;li&gt;start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not be after the current process date. &lt;li&gt;end date: Its the current process date. &lt;/ol&gt;</p> <p>@fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId , , It is the account id of the account for which statement period is to be fetched.</p> <p>@fcb.param MI,{@link com.ofss.fc.enumeration.account.AccountType},accountType, , It denotes the type of account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit.</p> <p>@return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons &lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt;</p> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteDeposit Details	@param accountId @return	This method will fetch the deposit details for the given account id and deposit number. @param accountId @param depositNumber @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteDeposit Details	@param accountId @return	This method will fetch the deposit details for the given account id and deposit number. @param accountId @param depositNumber @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteDeposit Details	@param accountId @return	This method will fetch the deposit details for the given account id and deposit number. @param accountId @param depositNumber @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchStatementAttributes	The method will fetch statement details for the given TD. @param accountId @param period @throws FatalException	This method will return all the details of a deposit that need to be displayed in the statement inquiry screen. @param sessionContext @param accountId @param depositNumber @param interestClass @param interestDivision @param effectiveDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchStatementAttributes	The method will fetch statement details for the given TD. @param accountId @param period @throws FatalException	This method will set the statement opening and closing balance for the current statement period. <p> The opening balance for system initiated request will be the lastStatementClosingBalance stored at the account level. If this value is null, meaning its the first statement that is generated, then the opening balance will be stored as zero. </br> In case of adhoc statements, opening balance will be the running balance of the last transaction that took place before the start date of the current statement request. </p> <p> The closing balance for both system and adhoc request will be simply the running balance of the last transaction in the statement pried of the current request. If there are no transactions in the period, then the closing balance will be equal to the opening balance. </p> @param statementRequest @param statementStatistics @param inquiryResponse @param depositDetails @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchStatementAttributes	The method will fetch statement details for the given TD. @param accountId @param period @throws FatalException	This method will set the statement opening and closing balance for the current statement period. <p> The opening balance for system initiated request will be the lastStatementClosingBalance stored at the account level. If this value is null, meaning its the first statement that is generated, then the opening balance will be stored as zero. </br> In case of adhoc statements, opening balance will be the running balance of the last transaction that took place before the start date of the current statement request. </p> <p> The closing balance for both system and adhoc request will be simply the running balance of the last transaction in the statement pried of the current request. If there are no transactions in the period, then the closing balance will be equal to the opening balance. </p> @param statementRequest @param statementStatistics @param inquiryResponse @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				depositDetails @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchTransactionFailure Logs		
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchTransactionFailure Logs		
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteAccount Details	@param accountId @return @throws FatalException	This method will place the next statement order. The method follows the below algorithm: <ol> <li>Check if the status of the account is either closed or Inactive. In each case ordering of next statement is not needed. <li>Calculate the beginning of the next statement period as end date of the last statement period plus one. <li>Calculate the end date of the next statement period using the statement frequency from Statement parameters. <li>Calculate the statement generation date with deferred statement generation date. Deferred generation date will only be taken into account if the frequency is monthly and above. Other wise statement generation date will be the same as the end date calculated. <li>Get the pricing ticket number from the pricing module. <li>Assemble a {@link TDStatementRequest} with all these details and call {@link StatementRequestService#orderStatement(IStatementRequest, String, String)} using the same. </ol> @param productStatementParametersDTO @param stmtGenTill @param termDepositAccountDTO @param nextStatementPeriodBeginDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchCompleteAccount Details	@param accountId @return @throws FatalException	This method will place the next statement order. The method follows the below algorithm: <ol> <li>Check if the status of the account is either closed or Inactive. In each case ordering of next statement is not needed. <li>Calculate the beginning of the next statement period as end date of the last statement period plus one. <li>Calculate the end date of the next statement period using the statement frequency from Statement parameters. <li>Calculate the statement generation date with deferred statement generation date. Deferred generation date will only be taken into account if the frequency is monthly and above. Other wise statement generation date will be the same as the end date calculated. <li>Get the pricing ticket number from the pricing module. <li>Assemble a {@link TDStatementRequest} with all these details and call {@link StatementRequestService#orderStatement(IStatementRequest, String, String)} using the same. </ol> @param productStatementParametersDTO @param stmtGenTill @param termDepositAccountDTO @param nextStatementPeriodBeginDate
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is minor and the operating instruction channel access type is no for given channel then return true,false otherwise @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will fetch the bank code for given account ID @param partyAccountRelationshipDetailsDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	fetchBankC ode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is operationally major and the operating instruction channel access type is no for given channel then return true,false otherwise @param partyAccountRelationshipDetailsDTO @return
IAccountTD Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountTDA dapter	isBackdated Transaction Performed OnAccount	This method determines whether back dated transaction is performed on Account. @param accountId @param accountTransferDate @return transactionPerformed	Method validateBackdatedTransactionsOnAccount is called to check whether there is any backdated transactions occurred on the account for the given posting date.   Since R2.2</br>
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAda pter	fetchParty WarningIn dicatorDeta ils	<p>This method returns the LoanAccountNotificationReponse that consist of various things like total arrears amount, advance payment, next due date etc. @param partyID @return the <tt>PartyNotificationResponse</tt> object. @throws FatalException	
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAda pter	fetchParty WarningIn dicatorDeta ils	<p>This method returns the LoanAccountNotificationReponse that consist of various things like total arrears amount, advance payment, next due date etc. @param partyID @return the <tt>PartyNotificationResponse</tt> object. @throws FatalException	<p>This method will fetch nomination details for a partyID and various details based on the party type. </p> @param partyID @return @throws FatalException
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAda pter	fetchAllAcc ountsOfPar ty	<p>This method returns the String[] of all the accountns ehich are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	This method is used to log notifications for a party/account for the first time and then fetch it. @param sessionContext @param partyId @param accountId @param logNotificationService @return @throws FatalException
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	<p>This method will call the BranchPartyAdapter to fetch all the accounts which are associated to that party. </p> @param partyID @return @throws FatalException
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	This method fetches notification logs. @param sessionContext @param partyId @param accountId @return @throws FatalException
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	This method fetches all accounts of a party. @param partyID @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAllAccountsOfParty	<p>This method returns the String[] of all the accounts which are associated with that party. @param partyID @return the <tt>String[]</tt> object. @throws FatalException	
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAccountNickName	<p>This method returns the Nick name of all the account which is associated with that party. @param partyID @param type @param accountID @return the <tt>String</tt> object. @throws FatalException	
IBranchParty Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchPartyAdapter	fetchAccountNickName	<p>This method returns the Nick name of all the account which is associated with that party. @param partyID @param type @param accountID @return the <tt>String</tt> object. @throws FatalException	
IServicePMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\IServicePMUAdapter	isEffectiveBundleExists	This method is used to get the status that whether there are effective bundles exist in the system currently or not. @throws FatalException	This method is used to get the status that whether there are effective bundles exist in the system currently or not.
IServicePMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\IServicePMUAdapter	fetchPartyBankPolicy		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServicePMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServicePMUAdapter	fetchDDABankPolicy		
ICardLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMAAdapter	inquireCreditFacility	This method is used to inquire credit facility @param sessionContext @param facilityId @return CreditFacilityDTO @throws FatalException	Fetch credit facility @param sessionContext @param facilityId @return @throws FatalException
ICardLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMAAdapter	inquireCreditFacility	This method is used to inquire credit facility @param sessionContext @param facilityId @return CreditFacilityDTO @throws FatalException	Populate Credit Card Details Post Inquiry @param sessionContext @param creditCardDetailsDTO @return @throws FatalException
ICardLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMAAdapter	inquireCreditFacility	This method is used to inquire credit facility @param sessionContext @param facilityId @return CreditFacilityDTO @throws FatalException	Link Add on card holder @param sessionContext @param addOnCardHolderDTOs @return @throws FatalException
ICardLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMAAdapter	inquireCreditFacility	This method is used to inquire credit facility @param sessionContext @param facilityId @return CreditFacilityDTO @throws FatalException	Fetch credit facility @param sessionContext @param facilityId @return @throws FatalException
ICardLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMAAdapter	inquireCreditFacility	This method is used to inquire credit facility @param sessionContext @param facilityId @return CreditFacilityDTO @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	amendFacil ityAmount	This method is used to amend the facility amount @param sessionContext @param facilityId @param sanctionedAmount @throws FatalException	The method changeCardLimit is used to change the limit for the credit card account.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, ,This attribute denotes the card account number. @fcb.param MI,MoneyDTO, requestLimit, ,This attribute denotes the request limit. @return {@link TransactionStatus} This class returns the transaction status @throws FatalException
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	amendFacil ityAmount	This method is used to amend the facility amount @param sessionContext @param facilityId @param sanctionedAmount @throws FatalException	
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	createFacilit y	This service will update the information for the facilities passed. @param sessionContext @param creditFacilityDTO @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMA dapter	createFacility	This service will update the information for the facilities passed. @param sessionContext @param creditFacilityDTO @throws FatalException	
ICardLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMA dapter	createOrFetchBorrowingEntity	This service is used to create or fetch borrowing entity for the party @param sessionContext, SessionContext @param partyDataDTO, PartyDataDTO @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException	This service is used to create or fetch borrowing entity for the party @param sessionContext, SessionContext @param partyDataDTO, PartyDataDTO @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException
ICardLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardLCMA dapter	createOrFetchBorrowingEntity	This service is used to create or fetch borrowing entity for the party @param sessionContext, SessionContext @param partyDataDTO, PartyDataDTO @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	createOrFet chBorrowin gEntity	This service is used to create or fetch borrowing entity for the party @param sessionContext, SessionContext @param partyDataDTO, PartyDataDTO @return BorrowingEntityInquiryResponse, , Contains the borrowing entity information created @throws FatalException	
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	generateFac ilityId	This service will retrieve the next available facility identifier from LCM. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO, creditFacilityDTO, , Contains the credit facility fields such as requested amount and term for facility and purpose. @return String, facilityId, , Indicates the facility identifier generated. @throws FatalException	This method is used to create credit facility using the external credit card details @param sessionContext @param creditCardDetailsDTO @param borrowingEntityId @return CreditFacilityDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	generateFac ilityId	This service will retrieve the next available facility identifier from LCM. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditFacilityDTO, creditFacilityDTO, , Contains the credit facility fields such as requested amount and term for facility and purpose. @return String, facilityId, , Indicates the facility identifier generated. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	defaultFacil ityCategory Fields	This service is used for defaulting facility category related fields in credit facility.  In Origination, only basic fields such as requested amount and term for facility, purpose  and variant information such as employee benefits required, whether facility is low doc  and capital subvention scheme is being used for the facility.    Facility Category for the product group under which a facility is being setup is inquired  from PMU. Facility Category fields are further fetched using this facility category code  and fields are defaulted to the facility being setup. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategory, , Indicates the facility category code @return CreditFacilityDTO, Returns the credit facility dto populated with default category fields @throws FatalException	This method is used to create credit facility using the external credit card details @param sessionContext @param creditCardDetailsDTO @param borrowingEntityId @return CreditFacilityDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	defaultFacil ityCategory Fields	This service is used for defaulting facility category related fields in credit facility.  In Origination, only basic fields such as requested amount and term for facility, purpose  and variant information such as employee benefits required, whether facility is low doc  and capital subvention scheme is being used for the facility.    Facility Category for the product group under which a facility is being setup is inquired  from PMU. Facility Category fields are further fetched using this facility category code  and fields are defaulted to the facility being setup. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategory, , Indicates the facility category code @return CreditFacilityDTO, Returns the credit facility dto populated with default category fields @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	validateFaci lityCategor y	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException
ICardLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \card\ICardLCMAadapt er	validateFaci lityCategor y	This service validates if facility category exists. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, facilityCategoryId, , Indicates the facility category code @throws FatalException	
IAssetClassifi cationEventP rocessingAda pter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \ac\IAssetClassification EventProcessingAdapter	persistActiv ityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	
IJournalCAS AAdapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \je\IJournalCASAAdapt er	debitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	debitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	debitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	 This method is used to debit the casa accounts. To do this DDATransactionApplicationService  is called. The entries are converted to AccountTransactionDetailsDTO and the DDATransactionApplicationService  debits the account. The debited accounts are put in a map and returned. 
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	creditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	creditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingCreate (to Create accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	creditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	 This method is used to credit the casa accounts. To do this DDATransactionApplicationService  is called. The entries are converted to AccountTransactionDetailsDTO and the DDATransactionApplicationService  credits the account. The credited accounts are put in a map and returned. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	creditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	 This method is used to credit the casa accounts for a reverse service call. To do this DDATransactionApplicationService is called. The entries are converted to AccountTransactionDetailsDTO   and the DDATransactionApplicationService reverses the account. The credited reversed accounts are put in   a map and returned. 
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	reverseDebitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	reverseDebitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	reverseDebitEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	 This method is used to debit the casa accounts for a reverse service call. To do this DDATransactionApplicationService is called. The entries are converted to AccountTransactionDetailsDTO   and the DDATransactionApplicationService reverses the account. The debited reversed accounts are put in   a map and returned. 
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	reverseCreditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	reverseCreditEntry	@param sessionContext @param entries @param objectMap @return map @throws FatalException	accountingReverse (to Reverse accounting and CASA Entries) @param sessionContext @param entries @return @throws FatalException
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	fetchCASADetails	@param sessionContext @param accountId @return JournalBasicInquiryDetailsDTO	
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	fetchCASADetails	@param sessionContext @param accountId @return JournalBasicInquiryDetailsDTO	
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	fetchCASADetails	@param sessionContext @param accountId @return JournalBasicInquiryDetailsDTO	
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	fetchCASADetails	@param sessionContext @param accountId @return JournalBasicInquiryDetailsDTO	Validates CASA account. If the CASA account's user defined status is blocked then validation  is raised.If either account's user defined status is debit blocked and account/ledger flag is debit  or credit blocked and account/ledger flag is credit then validation is raised.  If the account is not regular then validation error is raised 
IJournalCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalCASAAdapter	isValidAccountCode	@param sessionContext @param accountCode @return boolean	validateData (validate multi entry input data and call to validate Gaap and Ledger through adapter)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILinkedAccountInfoFromTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ILinkedAccountInfoFromTDAdapter	fetchAllAccountsForPartyId		
ILinkedAccountInfoFromTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ILinkedAccountInfoFromTDAdapter	fetchAccount		
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyBasicDetails		
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchAllCreditAssessment		Method to get the Customer Risk Score.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchAllCreditAssessment		Method to get the Customer Probability of default.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchAllCreditAssessment		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchEmploymentHistory		Method to get the Employment Status.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchEmploymentHistory		
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		Method to get the Party Type.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		Method to get the Party Type.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		Method to get the Party class.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		Method to get the Party Age.
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		Method to get the Warning Indicators.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISecurityPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ISecurityPartyAdapter	fetchPartyDetails		
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralDetails	@param p_FeederCollateralWrapperDTO @throws FatalException	@param p_FeederCollateralWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralDetails	@param p_FeederCollateralWrapperDTO @throws FatalException	
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralPartyOwnership	@param feederCollateralPartyWrapperDTO @throws FatalException	@param feederCollateralPartyWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralPartyOwnership	@param feederCollateralPartyWrapperDTO @throws FatalException	
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateInsurance	@param feederInstrDtlsWrapperDTO @throws FatalException	@param feederInstrDtlsWrapperDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateInsurance	@param feederInstrDtlsWrapperDTO @throws FatalException	
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralCharges	@param p_FeederCollateralChargesWrapperDTO @throws FatalException	@param pFeederCollateralChargesWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralCharges	@param p_FeederCollateralChargesWrapperDTO @throws FatalException	@param p_FeederCollateralWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralCharges	@param p_FeederCollateralChargesWrapperDTO @throws FatalException	
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralEntity	@param p_FeederCollateralEntityWrapperDTO @throws FatalException	@param pFeederCollateralEntityWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralEntity	@param p_FeederCollateralEntityWrapperDTO @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralGurantorInformation	@param feederCollateralGurantorWrapperDTO @throws FatalException	@param feederCollateralGurantorWrapperDTO
ILCMCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collection\ILCMCollectionAdapter	updateCollateralGurantorInformation	@param feederCollateralGurantorWrapperDTO @throws FatalException	
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	placeCollateralLien		
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		<p>This service represents the addCollateral method of the Collateral Application Service.&lt;br&gt;This method is used for create/add a new collateral into the system by capturing the details like Collateral Type,Collateral Category,Collateral sub category,Owner Estimated Market Value,Market Value,Bank Value and Bank Value relied on etc.,&lt;br&gt; And this service will also captures the Address details and Ownership details of the collateral and supports the all types(Property,Investments,Automobile,FixedAssest,Cash&amp;Deposit and Guarantee etc.) of collateral's to create into the system.&lt;br&gt; System will automatically generates unique id for every newly creating collateral. &lt;br&gt; While creating a collateral, system will retrieves the details from the collateral category preferences(like Bank haircut(%),Categorization Haircut(%),Charge Required) defined for the specified collateral type and collateral category and calculates the Bank Value,Category value,Bank Value Relied on and Capitalization Value.&lt;br&gt; And for newly creating collateral De-dupe search also performs to show the similar collateral's existing in the system to the end user. &lt;br&gt; @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO TODO: Explain the collateralDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		<p>This service represents the updateCollateral method of the Collateral Application Service.&lt;br&gt;This method is used for updating an existing collateral into the system by capturing the details like Collateral Type,Collateral Category,Collateral sub category,Owner Estimated Market Value,Market Value,Bank Value and Bank Value relied on etc.,&lt;br&gt; And this service will also captures the Address details and ownership details of the collateral and supports the all types(Property,Investments,Automobile,FixedAssest,Cash&amp;Deposit and Guarantee etc.) of collateral's to update into the system.&lt;br&gt; &lt;br&gt; While updating a collateral, system will retrieves the details from the collateral category preferences(like Bank haircut(%),Categorization Haircut(%),Charge Required) defined for the specified collateral type and collateral category and calculates the Bank Value,Category value,Bank Value Relied on and Capitalization Value.&lt;br&gt; &lt;br&gt; @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		<p>This service represents the closeCollateral method of the Collateral Application Service.&lt;br&gt;This method is used for close an existing collateral from the system based on collateral id&lt;br&gt;This service will change the object status flag to 'C'.These collateral's will be remain InActive in the system&lt;br&gt; &lt;br&gt; @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO TODO: Explain the collateralDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		<p>This service represents the addOrUpdateCollateral method of the Collateral Application Service.&lt;br&gt;This method is used for create/ update multiple new collateral's into the system by capturing the details like Collateral Type,Collateral Category,Collateral sub category,Owner Estimated Market Value,Market Value,Bank Value and Bank Value relied on etc.,&lt;br&gt; And this service will also captures the Address details and ownership details of the collateral and supports the all types(Property,Investments,Automobile,FixedAssest,Cash&amp;Deposit and Guarantee etc.,) of collateral's to create into the system.&lt;br&gt; System will automatically generates unique id for every newly creating collateral. &lt;br&gt; While creating a collateral, system will retrieves the details from the collateral category preferences(like Bank haircut(%),Categorization</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Haircut(%),Charge Required) defined for the specified collateral type and collateral category and calculates the Bank Value,Category value,Bank Value Relied on and Capitalization Value.&lt;br&gt; And for newly creating collateral De-dupe search also performs to show the similar collateral's existing in the system to the end user. &lt;br&gt; @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @param {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO TODO: Explain the collateralDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	updateCollateralLien		This represents the updateCondition method of the entity. @param sessionContext @param chargeDTO
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	removeCollateralLien		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	fetchCollateralLiens		
ICollateralsTDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\ICollateralsTDAdapter	fetchCollateralLiens		This service represents the fetchCollateral method of the Collateral Application Service. This method is used for retrieve the details of an existing collateral from the system based on collateral id  And this service will also fetches details the Address details and Ownership details of the collateral and additional details of all types(Property,Investments,Automobile,FixedAssest,Cash&Deposit and Guarantee etc.,) of collateral's and categories from the system.    @param {@link com.ofss.fc.app.context.SessionContext} sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @param {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralDTO} collateralDTO TODO: Explain the collateralDTO above @return {@link com.ofss.fc.app.lcm.dto.collaterals.CollateralInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	fetchBucketLimit	Method to return casa limit buckets for OD,Overline,TOD @param sessionContext @return List<String> @throws FatalException	This method would return the list of buckets limits for OD,Overline,TOD persisted in workitem

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	fetchBucketLimit	Method to return casa limit buckets for OD,Overline,TOD @param sessionContext @return List<String> @throws FatalException	System will check if the amounts are greater than zero that means the limit is breached. Sms will have configuration if to ignore/authorize/reject the transaction. Following limits will be checked. <li>if(DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.Overdraft).amount > 0) then following values are set while adding as error using OverridableBusinessPolicy#addValidationError(ValidationError error) this.objectName = DDABalanceLimitBusinessPolicy; this.attributeName = Overdraft; this.attributeValue = DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.Overdraft).amount; this.methodName = validateODLimit(); this.applicableAttributes = applicableAttributes; this.errorCode = errorCode; this.errorMessage = The transaction uses the overdraft limit from the account.;</li>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	fetchBucketLimit	Method to return casa limit buckets for OD,Overline,TOD @param sessionContext @return List<String> @throws FatalException	System will check if the amounts are greater than zero that means the limit is breached. Sms will have configuration if to ignore/authorize/reject the transaction. Following limits will be checked. <li>if(DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.Overline).amount > 0) then following values are set while adding as error using OverridableBusinessPolicy#addValidationError(ValidationError error) this.objectName = DDABalanceLimitBusinessPolicy; this.attributeName = Overline; this.attributeValue = DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.Overline).amount; this.methodName = validateOverlineLimit(); this.errorCode = errorCode; this.errorMessage = The transaction uses the overline balance from the account.</li>
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	fetchBucketLimit	Method to return casa limit buckets for OD,Overline,TOD @param sessionContext @return List<String> @throws FatalException	System will check if the amounts are greater than zero that means the limit is breached. Sms will have configuration if to ignore/authorize/reject the transaction. Following limits will be checked. <li>if(DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.TemporaryOverdraft).amount > 0) then following values are set while adding the same as error. this.objectName = DDABalanceLimitBusinessPolicy; this.attributeName = TemporaryOverdraft; this.attributeValue = DebitResponse.fundsAcquisitionResponse.fundsAcquiredMap.get(AcquisitionType.TemporaryOverdraft).amount; this.methodName = validateTODLimit(); this.errorCode = errorCode; this.errorMessage = The transaction uses the temporary overdraft limit from the account.</li>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	fetchBucketLimit	Method to return casa limit buckets for OD,Overline,TOD @param sessionContext @return List<String> @throws FatalException	This method returns the workList adapter to be used in interacting by DDA
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	approveWorkItemForApprovedForClosure		This method returns the workList adapter to be used in interacting by DDA
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	approveWorkItemForApprovedForClosure		This method is used to close the account closure request, only after successfully closing the account. @param applicationContext @param closureRequestToBeClosed
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	addWorkItemForApprovedForClosure		This method returns the workList adapter to be used in interacting by DDA

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	addWorkItemForApprovedForClosure		<p>The method will be used for updating the closure request details. It will primarily be used to update the retention strategy by the relationship manager. The other details cannot be updated once the request is submitted. &lt;pre&gt;The method works as follows :</p> <ul style="list-style-type: none"> <li>Call the repository method {@link DDAClosureRepository#getInstance} to create an instance of the repository.</li> <li>Call the method {@link DDAClosureRepository#update} to update the closure request details.</li> <li>If the retention strategy is proceed with closure request, then the request needs to be sent to the BEPL worklist.</li> <li>ElsIf the retention strategy is reject closure request, then</li> <li>Restore the original status by calling the method {@link AccountStatusManagementService#undoMarkedClosureStatus}.</li> <li>Call the repository method {@link DDAClosureRepository#close} to close the closure request.</li> </ul> <p>Check the overridable conditions by calling the method {@link OverridableFinancialBusinessPolicy#validate}&amp;lt;/li&gt;&amp;lt;/ul&gt;&amp;lt;/pre&gt; @param closureRequest</p>
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	deleteWorkItemForTransactionReferenceNumber		
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	deleteWorkItemForTransactionReferenceNumber		<p>This method is used to delete the workitem entry against the giben transaction reference number @param transactionReferenceNumber @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAWorkListAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAWorkListAdapter	deleteWorkItemForTransactionReferenceNumber		This method returns the workList adapter to be used in interacting by DDA
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	fetchFeesLinkedToAPurpose	This service will be used to query the list of price definitions maintained for a adhoc fee purpose id, for a specific domain category. Different price definitions codes can be maintained for a fee purpose for different domain categories. @param sessionContext @param domainCategory @param feePurposeId @return @throws FatalException	This service will be used to query the list of price definitions maintained for a adhoc fee purpose id, for a specific domain category. Different price definitions codes can be maintained for a fee purpose for different domain categories. @param sessionContext @param domainCategory @param feePurposeId @return feesIds
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	fetchPrice	This service is used to inquire the details of a price definition code. @param sessionContext @param priceDefinitionId @param effectiveDate @return @throws FatalException	This service is used to inquire the details of a price definition code. @param sessionContext @param priceDefinitionId @param PriceDTO @return
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	generateTicket	<p>This method will generate the pricing ticket for bundle subscription when the subscription request is made online. </p> @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	generateTicket	<p>This method will generate the pricing ticket for bundle subscription when the subscription request is made online. </p> @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	generateTicket	<p>This method will generate the pricing ticket for bundle subscription when the subscription request is made online. </p> @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	createFeeReco	<p>This method will create the fee reco for bundle subscription when the subscription request is made online. </p> @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	createFeeReco	<p>This method will create the fee reco for bundle subscription when the subscription request is made online. </p> @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	createFeeReco	<p>This method will create the fee reco for bundle subscription when the subscription request is made online. </p> @param pricingContractContextDTO @param evaluationDate @param currencyCodes @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	createPricingPlan	<p>This method will create the pricing plan. </p> @param pricingContractContextDTO @param arrangementId @param evaluationDate @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	createPricingPlan	<p>This method will create the pricing plan. </p> @param pricingContractContextDTO @param arrangementId @param evaluationDate @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	fetchFeeForPriceTicket	<p>This method will fetch price feature dto given a price ticket number. </p> @param priceTicketNumber @param currencyCodes @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	updateFeeDetailsForTicketNumber	<p>This method will intimate pricing once the bundle subscription fee is charged. </p> @param priceTicketNumber @param eventCodes @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	updateFeeDetailsForTicketNumber	<p>This method will intimate pricing once the bundle subscription fee is charged. </p> <p>@param priceTicketNumber @param eventCodes @throws FatalException</p>	<p>This method is exposed to external modules to do fees management for CASA, Loan and settlement module. Based on the input, the fees will be debited and then intimated to pricing. In case fee details are not passed as input, then fees will be calculated, debited and intimated to pricing. Following are the required steps:</p> <ol style="list-style-type: none"> <li>Fetch pricing details</li> <li>Debit the fee from account.</li> <li>Raise accounting for the charged fees</li> <li>Intimate pricing</li> </ol> <ol style="list-style-type: none"> <li>Fetch pricing details If price ticket number is provided in input than inquire fees based on price ticket number. Price ticket number in input indicates that fees is already inquired by the caller. There could be negotiation performed. The final information is stored into the pricing against this price ticket number. else if pricing details are not provided then inquire fees by the event id. It is not know at this point that whether there is fee linked with this event or not. This call to pricing will return PD with amount, NSFTreatment, discounts and additional fee if there is any. If there is no fee then there will not be any further processing.</li> <li>Debit the fee from account. Debit account can be any module account like CASA, Loan, 2. This method will be responsible to debit the fee from account.</li> <li>Raise accounting for the charged fees</li> <li>Intimate the pricing system by calling AccountPricingAdapter.intimatePricing(priceTicketNumber, transactionReferenceNumber): boolean</li> </ol> <p>@throws FatalException @fcb.param I,String,priceTicketCode,40,, Dialogue ticket obtained from Pricing for fees applicable. If this is supplied, then wrappedPriceFeatureDTO is not required @fcb.param</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				I,WrappedPriceFeatureDTO,wrappedPriceFeatureDT O,,,Negotiated fees. If this is supplied, then priceTicketCode is not required @fcb.param I,String,arrivalId,40, , Indicates the entity for which fees are applicable during the ongoing event. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI, String, accountCode,40 ,, Indicates the account from which the fees have to be funded @fcb.param I, String, eventId, 40,, Indicates the ongoing event for which the fees are applicable on the entity. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,transactionDesc,255, , Transaction description to be logged on the funding account
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	maintainPollerStatistics	This method is used to update the {@link PollerStatistics} @param pollerStatisticsDTO @throws FatalException	This method is used to update the {@link PollerStatistics} @param pollerStatisticsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getPeriodicPricesForBundle	Method to get periodic prices for bundle reference id @param context @param bundleReferenceNo @param evaluationDate @param computationDate @param partyId @param accountCurrency @return @throws FatalException	<p>This method will be used to fetch the next charging date for bundle maintenance fee. When bundle subscription is done online, the computation date for periodic fee will be available and {@linkplain com.ofss.fc.app.pricing.service.contract.EventPriceApplicationService.getPeriodicPricesWithPeriodicDates} service will be called to fetch the next charging date. However, when bundle subscription request is made via Origination, the computation date is not available and {@linkplain com.ofss.fc.app.pricing.service.contract.PeriodicFeeSimulationApplicationService.computePeriodicFeeDates} service will be called to fetch the next charging date. </p> @param partyBundleLinkage
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	intimateFee waiver	<p>This method will send an intimation to pricing when a fee is waived. </p> @param domain @param domainCategory @param factContextDTO @param isAdhocEvent @param transactionNature @param transactionDetailsDTOs @return	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getWrapperPriceFeatureDTOForPriceTicket	<p>This method will fetch the fee details given the price ticket number. </p> @param currencyCodes @param pricingTicketNumber @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getWrappedPriceFeatureDTOForPriceTicket	<p>This method will fetch the fee details given the price ticket number. </p> @param currencyCodes @param pricingTicketNumber @return @throws FatalException	<p>This method will be used to charge the bundle subscription fee when the bundle subscription is happening online.  If the user chooses ChargeActionType.COLLECT, this method will call the fee management service to do fee processing, intimate pricing and return the amount charged. If the user chooses ChargeActionType.WAIVE, this method will not debit the account, it will only intimate pricing and return the amount charged as ZERO. </p> @param partyBundleLinkage @param nominatedAccountId @param nominatedAccountModule @param nominatedAccountCurrency @param nominatedAccountProductCode @param nominatedAccountOfferId @param nominatedAccountBranchCode @param eventId @param wrappedPriceFeatureDTOArray @param componentPreferenceDTO @return @throws FatalException
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getWrappedPriceFeatureDTOForPriceTicket	<p>This method will fetch the fee details given the price ticket number. </p> @param currencyCodes @param pricingTicketNumber @return @throws FatalException	This service is used to compute and return the PriceFeatureDTO. The service is called based on priceTicketNumber. The service internally calls the pricing service {@link EventPriceApplicationService#getFeesForTicketNumber} @param priceTicketCode @param cyyCode @throws FatalException
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getWrappedPriceFeatureDTOForPriceTicket	<p>This method will fetch the fee details given the price ticket number. </p> @param currencyCodes @param pricingTicketNumber @return @throws FatalException	This method will fetch all the fees for the given ticket number. For the fees that are marked as collected, it will raise an accounting event to refund the amount in suspense GL @param applicationContext @param pricingTicketNo @param currency @param domainCategory @param event @param arrangementNo @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getComputedPeriodicFeeDate	<p>This method will fetch the next charge dates for the given arrangement number. </p> @param pricingContextDTO @param arrangementNumber @param startDateForPeriodicCycle @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getComputedPeriodicFeeDate	<p>This method will fetch the next charge dates for the given arrangement number. </p> @param pricingContextDTO @param arrangementNumber @param startDateForPeriodicCycle @return @throws FatalException	<p>This method will be used to fetch the next charging date for bundle maintenance fee. When bundle subscription is done online, the computation date for periodic fee will be available and {@linkplain com.ofss.fc.app.pricing.service.contract.EventPriceApplicationService.getPeriodicPricesWithPeriodicDates} service will be called to fetch the next charging date. However, when bundle subscription request is made via Origination, the computation date is not available and {@linkplain com.ofss.fc.app.pricing.service.contract.PeriodicFeeSimulationApplicationService.computePeriodicFeeDates} service will be called to fetch the next charging date. </p> @param partyBundleLinkage
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getPriceOnEvent	@param sessionContext @param pricingEventContextDTO @param accountId @param eventId @param processDate @param currencyCodes @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getPriceOnEvent	@param sessionContext @param pricingEventContextDTO @param accountId @param eventId @param processDate @param currencyCodes @return	<p>This method calculates the fees for a given event. This method internally calls the Pricing module's  {@link EventPriceApplicationService#getPriceOnEvent()} via adapter to actually calculate the fees and return the same back to the caller. For a particular event, multiple fees may be calculated with the same pricing ticket number. Hence an array of {@link PriceFeatureDTO} is returned back to the caller method.    The following is the algorithm for the method:    Arrangement Id, Event Id, Arrangement Module, Task Module Type, Process Date, cyyCode array, FactDTO is passed as input parameter to this method. Get the AccountPricingAdapter by calling {@link getAccountPricingAdapterInstance(): AccountPricingAdapter} / </p> <p> Call {@link IAccountPricingAdapter#getFeeOnEvent()} passing the required inputs and PricingDetails will be returned from Pricing by this call through adapter.</p> @param arrangementId String @param eventId String @param arrangementModule TaskModuleType @param processDate @param cyyCode String[] @param factDTO @throws FatalException
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	intimatePricing	@param pricingContext @param transactionNature @param transactionDetailsDTOS @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPricingAdapter	intimatePricing	@param pricingContext @param transactionNature @param transactionDetailsDTOS @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	intimatePricing	@param pricingContext @param transactionNature @param transactionDetailsDTOS @return @throws FatalException	<p>&lt;p&gt;This method intimates the Pricing module of the charged fee details in the CASA, Loan and Settlement mode module. &lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;br&gt; This is the context in which this method will be called.&lt;br&gt; &lt;br&gt; For a given event, CASA, Loan and Settlement mode module will call pricing to calculate the fees to be charged.&lt;br&gt; &lt;br&gt; &lt;br&gt; Pricing will calculate the fees based on Pricing plan and return the same.&lt;br&gt; &lt;br&gt; &lt;br&gt; CASA, Loan and Settlement mode module has the option to charge the fees, waive them or apply them as it is.&lt;br&gt; &lt;br&gt; &lt;br&gt; The actual action taken on the fees needs to be communicated back to pricing.&lt;br&gt; &lt;br&gt; &lt;br&gt; The details of the fees are passed back to pricing via this method to be updated in the various buckets against the pricing ticket number.&lt;br&gt; &lt;/p&gt; &lt;br&gt; &lt;p&gt; Following are the required steps:&lt;br&gt; 1. get the instance of account accounting adapter &lt;br&gt; {@link com.ofss.fc.app.adapter.impl.account.AccountPricingAdapter}&lt;br&gt; 2. call intimate pricing of adapter &lt;br&gt; {@link com.ofss.fc.app.adapter.impl.account.AccountPricingAdapter#intimatePricing} &lt;/p&gt; @param priceTicketNumber @param transactionReferenceNumber @param taskModuleType @param accountId @param chargedFeeNode @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	intimatePricing	@param pricingContext @param transactionNature @param transactionDetailsDTOS @return @throws FatalException	<p>&lt;p&gt;This method is exposed to external modules to do fees management for CASA,Loan and settlement module.&lt;br&gt; Based on the input, the fees will be debited and then intimated to pricing. &lt;br&gt; In case fee details are not passed as input, then fees will be calculated, debited and intimated to pricing.&lt;br&gt; Following are the required steps:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Fetch pricing details &lt;br&gt; &lt;li&gt;Debit the fee from account.&lt;br&gt; &lt;li&gt;Raise accounting for the charged fees&lt;br&gt; &lt;li&gt;Intimate pricing &lt;br&gt; &lt;/ol&gt; &lt;ol&gt; &lt;li&gt;Fetch pricing details If price ticket number is provided in inupt than inquire fees based on price ticket number.Price ticket number in input indicates that fees is already inquired by the caller. There could be negotiation performed. The final information is stored into the pricing against this price ticket number. else if pricing detials are not provided then inquire fees by the event id. It is not know at this point that whether there is fee linked with this event or not. This call to pricing will return PD with amount, NSFTreatment, discounts and additional fee if there is any. If there is no fee then there will not be any further processing. &lt;br&gt; &lt;li&gt;Debit the fee from account. Debit account can be any module account like CASA, Loan, 2. This method will be responsible to debit the fee from account. &lt;br&gt; &lt;li&gt;Raise accounting for the charged fees&lt;br&gt; &lt;li&gt;Intimate the pricing system by calling AccountPricingAdapter.intimatePricing(priceTicketNumber,transactionReferenceNumber): boolean &lt;/ol&gt;&lt;/p&gt; <p>@throws FatalException @fcb.param I,String,priceTicketCode,40,, Dialogue ticket obtained from Pricing for fees applicable. If this is supplied, then wrappedPriceFeatureDTO is not required @fcb.param</p> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				I,WrappedPriceFeatureDTO,wrappedPriceFeatureDTO,,,Negotiated fees. If this is supplied, then priceTicketCode is not required @fcb.param I,String,arrivalId,40, , Indicates the entity for which fees are applicable during the ongoing event. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI, String, accountCode,40 ,, Indicates the account from which the fees have to be funded @fcb.param I, String, eventId, 40,, Indicates the ongoing event for which the fees are applicable on the entity. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,transactionDesc,255, , Transaction description to be logged on the funding account
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	markFeeAsReversed	This service will mark the the fee as reversed @param domainCategory @param ticketNumber @param eventCode @throws FatalException	This method will fetch all the fees for the given ticket number. For the fees that are marked as collected, it will raise an accounting event to refund the amount in suspense GL @param applicationContext @param pricingTicketNo @param currency @param domainCategory @param event @param arrangementNo @throws FatalException
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	fetchFromArrangementNumberForGivenDate	@param arrangementNumber @param asOnDate @return	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	fetchFromArrangementNumberForGivenDate	@param arrangementNumber @param asOnDate @return	<p>This method populates the BundleAccountLinkage entity. </p> @param bundleSubscriptionDTO @param bundleReferenceId @param accountBasicDetailsDTOArray @return bundleAccountLinkageArray

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getDeferredFeeWithCurrentCycleFee	@param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param evaluationDate @param computationDate @param currencyCodes @param periodFactInformationDTOArray @param chargeActionTypeArray @return @throws FatalException	<p>This method will inquire the prices to be charged on bundle maintenance fee charging date when there is no nominated account to collect the fee from. It will inquire all the fees and call fee management service to update the fee as derffered. </p> @param applicationContext @param partyBundleLinkage
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getDeferredFeeWithCurrentCycleFee	@param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param evaluationDate @param computationDate @param currencyCodes @param periodFactInformationDTOArray @param chargeActionTypeArray @return @throws FatalException	<p>This method will inquire the prices to be charged on bundle maintenance fee charging date when there is no nominated account to collect the fee from. It will inquire all the fees and call fee management service to update the fee as deferred. </p> @param applicationContext @param partyBundleLinkage
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getDefferedFee	@param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param currencyCodes @return @throws FatalException	@param applicationContext @param partyBundleLinkage @param newNominatedBundleAccountLinkage @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getChargedFeesForDeferment	This method is used to fetch already charged periodic fees between back date and evaluation date.This charged fees are send out to the calling module as Reversal Details DTO to mark this fees as reversed and new price feature is populated and send out to the calling module as DeferPriceFeature to mark this fees as Defer. @param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param backDate @param asOnDate @param currencyCodes @return @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getChargedFeesForDeferment	This method is used to fetch already charged periodic fees between back date and evaluation date.This charged fees are send out to the calling module as Reversal Details DTO to mark this fees as reversed and new price feature is populated and send out to the calling module as DeferPriceFeature to mark this fees as Defer. @param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param backDate @param asOnDate @param currencyCodes @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	updateBundleVarianceExpiryDate	This method will update all bundle variance end date for a given arrangementNumber.   Currently it updates only Price variances. @param arrangementNumber The arrangementNumber @param asOnDate The asOnDate @param bundleVarianceExpiryDate The bundleExpiryDate @return TransactionStatus @throws FatalException	
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	updateBundleVarianceExpiryDate	This method will update all bundle variance end date for a given arrangementNumber.   Currently it updates only Price variances. @param arrangementNumber The arrangementNumber @param asOnDate The asOnDate @param bundleVarianceExpiryDate The bundleExpiryDate @return TransactionStatus @throws FatalException	This method notifies the linkage of an account to bundle. @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getPriceForAccountClosure	This method will be called to get all the details of prices during account closure. @param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param currencyCodes @param asOnDate @param accountClosingDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPricingAdapter	getPriceForAccountClosure	This method will be called to get all the details of prices during account closure. @param arrangementCode @param domainCategory @param priceTicketNumber @param factContextDTO @param currencyCodes @param asOnDate @param accountClosingDate @return @throws FatalException	@param partyBundleLinkage @return
IBankDraftReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IBankDraftReportAdapter	printBankDraft		
IBankDraftReportAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IBankDraftReportAdapter	printBankDraft		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductDetails	This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired @return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws	This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for this pricing plan.  This reference Id is generated using the account module and is generated only once  for the given submission.  2. The Pricing Plan is created using this bundle reference Id.   3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.  4. Pricing Module is called to get pricing information for this Bundle Subscription Event.  5. If bundle

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>           FatalException &lt;br&gt; <a href="#">{@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND}</a>&lt;br&gt;           FC_OR_DE_0054 error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt; </p>	<p>           subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.&lt;br&gt; 6. Else Fees can be negotiated, waived or deferred for this bundle reference Id. &lt;br&gt; The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.&lt;br&gt;           @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO, ,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkageDTO, ,Product Group information for the group nominated for bundle subscription fees. @return <a href="#">{@link PricingResponse}</a> This class holds ticket number for bundle fees which is stored.&lt;br&gt; All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException         </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductDetails	<p>This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired</p> <p>@return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND}&lt;br&gt; FC_OR_DE_0054 error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductDetails	<p>This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired</p> <p>@return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND}&lt;br&gt; FC_OR_DE_0054 error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductDetails	<p>This service is used for inquiring the details for the product code provided @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product for which the details are being inquired</p> <p>@return {@link ProductInquiryResponse}, This class represents the response containing the detailed information about the product being inquired @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND}&lt;br&gt; FC_OR_DE_0054 error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios. &lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt; &lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param insuranceType @param productGroupCode</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This method would perform the following operations</p> <ul style="list-style-type: none"> <li>PMU --&gt; fetch Product Group Details On BusinessUnit&lt;/li&gt;</li> <li>PMU --&gt; fetch Credit Policy Details&lt;/li&gt;</li> </ul> <p>The Credit Policy Template DTO is then fetched from the Policy Details Response and returned to the invoker @param sessionContext Session Context Object @param productGroupId Product Group ID @param businessUnit Business Unit @return Credit Policy Template DTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios. &lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt; &lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param insuranceType @param productGroupCode</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType &lt;br&gt; LRM required is obtained from Origination Preference. Lending Product Group Linkage related facts are &lt;br&gt; sent to the rule engine to execute rule for that rule Id whose response will be LRM Required or not. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information &lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This service will perform the following validations:&lt;br&gt; 1&gt; Fetch details from the credit policy template maintained for the product group.&lt;br&gt; Check if internal accounts to be created is mandatory and verify if atleast one account&lt;br&gt; has been configured in such cases.&lt;br&gt; 2&gt; Check for LVR and LTV code deteriorations.&lt;br&gt; 3&gt; Execute rules for LRM policy and update the product group details for LMI flag applicability.&lt;br&gt; 4&gt; Check if at least one account is configured and nominated for capitalization if Capitalization of fees is selected for the product group. Also more than one accounts cannot be nominated for capitalization.&lt;br&gt; 5&gt; During post assessment structure solution, if total requested amount (including capitalized amount) for the main line is greater than the approved amount with cushion then force user to edit submission&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO, , Indicates the lending product group being validated @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>Calculate number of days for account opening or first disbursement based on the product group / credit policy details @param sessionContext @param productGroupId @param facilityId @param productGroupLinkageType @param creditFacility @return accountOpeningOrFirstDisbursementDays @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This method creates an array of {@link ProductDetailsDTO} which contains information about the products which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @param creditFacilityDTOArray the facility hierarchy which includes the parent and its child facilities list of {@link CreditFacilityDTO} @return an array of {@link ProductDetailsDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupId, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed&lt;br&gt; &lt;ol&gt; &lt;li&gt;Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit&lt;/li&gt; &lt;li&gt;Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param productGroupId @param businessUnit @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This method validate the Lending Product Group requested term exceeding the maximum or minimum. @param sessionContext @param submissionInputDTO @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	<p>This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group</p> <p>@return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}&lt;br&gt; FC_OR_DE_0059 error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt;</p>	<p>This service will add error for each warning indicator. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String, partyId, , Indicates the party for which applicant warnings are being checked. @fcb.param MI,String, eventCode, , Indicates the event for which warning indicators are being checked. @fcb.param MI,ModuleType, moduleType, , Indicates the event is being fetched for which Product group. @fcb.param MI,String, productGroupId, , Indicates the product group identifier @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetails	This service will fetch the product group details for the product group identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group @return ProductGroupInquiryResponse, Indicates the response containing detailed product group information. @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed  <ol> <li>Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit</li> <li>Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId</li> </ol> @param productGroupId @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductInProductGroup	This service will check if the product is valid in the product group i.e. it belongs to that product group. @fcb.param MI, String, productCode, , Indicates the identifier for the product @fcb.param MI, String, productGroupCode, , Indicates the identifier for the product group @return boolean, Indicates if the product is valid and belongs to the product group @throws	This method validates whether the given product code is in the given product group or not. The method returns "true" if the product belongs to the group The method returns "false" if the product is not in the group. @param productCode String, productGroupCode String @return productInGroup boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validatePurposeType	This service will validate purpose type in product Group . @param productGroupCode @param purposeType @return @throws FatalException   {@link OriginationErrorConstants#PURPOSE_NOT_FOUND}  FC_OR_DE_0013 error occurs when:  The purpose selected for the product group is not present in the allowed list of purposes for that product group.  {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	checks if ProductGroup and Respective Purpose Type Exists @param fundingTableTemplateDTO
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	checks if ProductGroup and Respective Purpose Type Exists @param fundingTableTemplateDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This method would perform following validations <ul> <li>Check for offer expiry for OD, TD, DD and LN offers present in the submission</li> <li>Check for Back dated account opening</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This method is called from the service validateSubmission. It validates whether all the product groups present in the submission are valid one and the requested amount and requested terms are in the allowed range for the product group. If any validation fails it should add an exception to validation error with proper error message @param submissionInputDTO @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group	This service performs null checks for defaulting product group fields to facility @param productGroupLinkageDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			identifier is incorrect.   	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This service performs null checks for defaulting product group fields to facility @param productGroupLinkageDTO @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This method is called from the service validateSubmission. It validates whether all the product groups present in the submission are valid one and the requested amount and requested terms are in the allowed range for the product group. If any validation fails it should add an exception to validation error with proper error message @param submissionInputDTO @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateProductGroup	This service will validate productgroup code. @param productGroup @return @throws FatalException   {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND}  FC_OR_DE_0059 error occurs when:  The details for the product group identifier cannot be fetched or the product group identifier is incorrect.   	This service will mark the input product group as withdrawn or rejected. If the whole submission has been withdrawn, all product groups within the submission will be withdrawn.  If a single product group is rejected (by rejecting any document at product group level) or application is rejected (by rejecting the offer letter), the product group state is marked as REJECTED.   Facility details are deleted in both the above cases  If the product group is a part of the bundle, a bundle exception log is generated and saved.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ProductGroupLinkageDTO productGroupLinkageDTO, , Indicates the product group being withdrawn or rejected. @fcb.param MI,String event,Indicates the corresponding event set by the caller @fcb.param CI,String nominatedOfferId, , Indicates the offer identifier for the product group nominated for bundle fees. This is applicable only if product group is part of a bundle @fcb.param CI,String applicationBranch, ,Indicates the application branch. This is applicable only if product group is part of a bundle @fcb.param OI,String withDrawComments, , Indicates the reason for withdrawal of product group. This is applicable only if product group is part of a bundle @fcb.param MI,boolean isMarkedForExpiry,Indicates the submission is marked for expiry @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCASAPreferencesInquiryResponse	This service will fetch the casa preferences for the casa product identifier passed. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, productCode, , Indicates the identifier for the product @return CasapreferencesDTO, Indicates the CASA preferences for the product code passed.	@param sessionContext @param bundleOfferReferenceDTO @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchEligibleOffersForProductGroup	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws FatalException	This service is used to fetch the eligible offers for the selected product group from PMU module.  An eligibility check is run based on application channel, applicants applying for the product group and facility parameters.  A list of offers complying with the above conditions are returned. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantDataDTO[], applicantDataDTOs, , Contains information about applicants applying for the product group. @fcb.param MI, ProductDetailsDTO, , Contains information about product group and facility in case of lending product groups. @fcb.param MI, HeaderDTO, submissionHeaderDTO, , Indicates the branch details, channel details etc for a submission @return OfferInquiryResponse, Returns list of eligible offers for the product group. @throws FatalException @exception   {@link OriginationErrorConstants#CAMPAIGN_CODE_INV

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ALID} : FC_OR_DE_0085   This exception is thrown if campaign code provided is incorrect or not valid for the product group.     {@link OriginationErrorConstants#FAILED_IN_RETRIEVING_ELIGIBLE_OFFERS} : FC_OR_DE_0086   This exception is thrown if offers cannot be retrieved for the product group.     {@link OriginationErrorConstants#OR_REQUESTED_TERM_NULL} : FC_OR_DE_0101   This exception is thrown if the requested term is null.  
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchEligibleOffersForProductGroup	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws FatalException	This method will populate all facts such as Facility, Party etc. and make an inquiry from PMU to get all eligible offers. @param sessionContext @param mainPFLinkage @param applicationResponse @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchEligibleOffersForProductGroup	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchEligibleOffersForProductGroup	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios. &lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt; &lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param insuranceType @param productGroupCode</p>
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		<p>This method would perform the following operations &lt;ul&gt; &lt;li&gt;PMU --&gt; fetch Product Group Details On BusinessUnit&lt;/li&gt; &lt;li&gt;PMU --&gt; fetch Credit Policy Details&lt;/li&gt; &lt;/ul&gt; The Credit Policy Template DTO is then fetched from the Policy Details Response and returned to the invoker @param sessionContext Session Context Object @param productGroupId Product Group ID @param businessUnit Business Unit @return Credit Policy Template DTO @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		<p>&lt;ol&gt;This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI&lt;br&gt;This method will be used to Fetch all possible CCI policies based on following different scenarios. &lt;li&gt;If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.&lt;/li&gt; &lt;li&gt;If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of response.&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param insuranceType @param productGroupCode</p>
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		<p>This method needs to be called if LRM is applicable at the product group level. The rule Id for RuleType &lt;br&gt;LRM required is obtained from Origination Preference. Lending Product Group Linkage related facts are &lt;br&gt; sent to the rule engine to execute rule for that rule Id whose response will be LRM Required or not. @fcb.param MI SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO , This stores information &lt;br&gt; regarding the product group such as requested amount, requested tenor, LenderRiskMgmtDetailsDTO, etc @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		This service will perform the following validations:  1> Fetch details from the credit policy template maintained for the product group.  Check if internal accounts to be created is mandatory and verify if atleast one account  has been configured in such cases.  2> Check for LVR and LTV code deteriorations.  3> Execute rules for LRM policy and update the product group details for LMI flag applicability.  4> Check if at least one account is configured and nominated for capitalization if Capitalization of fees is selected for the product group. Also more than one accounts cannot be nominated for capitalization.  5> During post assessment structure solution, if total requested amount (including capitalized amount) for the main line is greater than the approved amount with cushion then force user to edit submission  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,LendingProductGroupLinkageDTO lendingProductGroupLinkageDTO, , Indicates the lending product group being validated @fcb.param MI,CreditFacilityDTO[] creditFacilityDTOs, , Indicates the facility details. @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		Calculate number of days for account opening or first disbursement based on the product group / credit policy details @param sessionContext @param productGroupId @param facilityId @param productGroupLinkageType @param creditFacility @return accountOpeningOrFirstDisbursementDays @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		This method creates an array of {@link ProductDetailsDTO} which contains information about the products which are linked to the given facilities @param sessionContext the {@link SessionException} @param parentCreditFacilityDTO the parent facility {@link CreditFacilityDTO} @param creditFacilityDTOArray the facility hierarchy which includes the parent and its child facilities list of {@link CreditFacilityDTO} @return an array of {@link ProductDetailsDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed  <ol> <li>Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit</li> <li>Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId</li> </ol> @param sessionContext @param productGroupId @param businessUnit @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchProductGroupDetailsOnBusinessUnit		This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed  <ol> <li>Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit</li> <li>Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId</li> </ol> @param productGroupId @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	checkIfOffersValid	This service check if the offer passed is valid and not expired. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String offerId, , Indicates the offer identifier being validated @return Boolean, if offer is valid and not expired. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	checkIfOffersValid	This service check if the offer passed is valid and not expired. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String offerId, , Indicates the offer identifier being validated @return Boolean, if offer is valid and not expired. @throws FatalException	@param productOfferReferenceDTOs
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSubmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This service will inquire the application details for the application identifier passed.   This inquiry is to fetch the basic details of the application such as application channel, branch, product groups attached, submission to which application is linked,current application status , application type and the applicants that are attached to the application.   It will also update the agent details in header if the submission was sourced by agent.  In addition, simulation details attached to product groups for a given application are also inquired in this service.  In case application is for Amend, the amend details will be populated with amend actions performed and the original values.   @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String, applicationId, , Indicates the identifier for the application. @return ApplicationInquiryResponse, An instance of the response containing basic application details and applicant information linked to the application. @throws FatalException @exception    

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				{@link OriginationErrorConstants#APPLICATION_NOT_FOUND} : FC_OR_DE_0052   This error occurs when:  The application identifier is invalid and no application details are found for the same. 
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSubmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This service will cancel existing bundle already linked to a submission. It will also be called during update of bundle detail. If new bundle reference id is same as existing bundle reference id then bundle will not be cancelled.  This service will perform the following actions-   i) Check if any bundle is linked with submission   ii) If yes, check if different bundle has been selected or user has cancelled bundle  iii)If existing bundle is present, move the bundle details to bundle history   iv) Delete bundle details from main tables   v) Intimate bundle module by logging failure exception for unfulfilled offers linked to bundle @param sessionContext @param submissionId @param bundleReferenceId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This service is used for displaying the Submission Summary in the Application Tracker.  It is used for displaying details for all product groups (lending as well as deposit).  All applicants applying to the submission are also displayed.  In case of lending application, facility details , collateral details and funding details are also present.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,submissionId, The id of the submission whose summary is being viewed. @return An instance of {@link com.ofss.fc.app.origination.dto.core.submission.DraftSubmissionInquiryResponse} Contains the Submission Input details for submission Id passed. @throws FatalException @exception   {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055   This error occurs when:  The details for the submission identifier were not found or the submission identifier is incorrect. 
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This method will perform the following-   i) Fetch existing bundleDetails.  ii) For offers whose accounts have already been opened or which have been withdrawn, the flag isOfferProcessed will be set to True.   iii) Set updated bundleDetailsDTO in submissionInputDTO. @param submissionInputDTO
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This method gets called when the withdraw Submission is selected. This service will do the following operations  <li>1. Validates for whether any account is opened under this submission. If no account gets opened, it can proceed further, otherwise throw proper error message</li>  <li>2. Change the status of all the Applications under this submission to Withdrawn</li>  <li>3. Close all the product groups under each application.</li>  <li>4. If the product group is of <li>3. Delete all the facilities linked part of this submission</li>   <li>4. Kill all the process which are initiated by this submission</li>   <li>5. Finally change the status of Submission to Withdrawn</li>  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, WithdrawSubmissionDetailsDTO, withdrawSubmissionDetailsDTO, , Indicates the instance of WithdrawSubmissionDetailsDTO. @fcb.param MI, String, submissionId, , Indicates the identifier for the submission. @fcb.param MI,Boolean isSubmissionFeeNegotiated,,It is true when withdraw



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>fees is negotiated or waived and further requires approval for WithdrawSubmission Task. @fcb.param MI, SubmissionApprovalDataDTO submissionApprovalDataDTO, Indicates the submission level Approval Data DTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#OR_APPLICATION_WITHDRAWN_NOT_ALLOWED} : FC_OR_WA_0001 &lt;br&gt; This error occurs when:&lt;br&gt; The application under the submission is already withdrawn or processed. Such submissions cannot be withdrawn if any of the applications under it are already processed.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_ALREADY_WITHDRAWN} : FC_OR_WA_0002 &lt;br&gt; This error occurs when:&lt;br&gt; A submission is already withdrawn and the user tries to withdraw same submission again. This will happen when the same submission is being accessed through human task and application tracker.&lt;br&gt; The user inquires the submission through application tracker and then withdraws the submission from the human task and later returns to the tracker to withdraw.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_WITHDRAWN_NOT_ALLOWED} : FC_OR_WA_0004 &lt;br&gt; This error occurs when:&lt;br&gt; Accounts have</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				been opened under an application and the user tries to withdraw such a submission.     {@link OriginationErrorConstants#SUBMISSION_CANNOT_WITHDRAWN} : FC_OR_WA_0005   This error occurs when:  Accounts have been opened under an application and the user tries to withdraw such a submission.  
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchBundleDetails	This method is called from the service validateSibmission.  It will fetch the bundle information for the given bundle id. @param sessionContext @param bundleId @return @throws FatalException	This method is called from validateSubmission. If the submission is of product bundle type then it validates whether all the mandatory product groups for the product bundle is applied or not. If the validation fails it should add an exception to validation error with proper error message @param submissionInputDTO @param bundleId @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		<ol>This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI This method will be used to Fetch all possible CCI policies based on following different scenarios. <li>If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.</li> <li>If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of respnse.</li> </ol> @param sessionContext @param insuranceType @param productGroupCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		This method would perform the following operations <ul style="list-style-type: none"> <li>PMU --&gt; fetch Product Group Details On BusinessUnit</li> <li>PMU --&gt; fetch Credit Policy Details</li> </ul> The Credit Policy Template DTO is then fetched from the Policy Details Response and returned to the invoker @param sessionContext Session Context Object @param productGroupId Product Group ID @param businessUnit Business Unit @return Credit Policy Template DTO @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		<ol style="list-style-type: none"> <li>This method will be used to show all possible CCI policy applicant can opt from, if it decides to opt for CCI</li> <li>This method will be used to Fetch all possible CCI policies based on following different scenarios. <ul style="list-style-type: none"> <li>If there are no policies maintained/linked for given product Group and Credit Policy template then all policies fetch from Insurance would be sent back.</li> <li>If there are policies maintained/lined for given product group and credit policy template then only those will be sent back as part of respnse.</li> </ul> </li> </ol> @param sessionContext @param insuranceType @param productGroupCode
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		@param sessionContext @param inquiryResponse @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed  <ol> <li>Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit</li> <li>Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId</li> </ol> @param sessionContext @param productGroupId @param businessUnit @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCreditPolicyDetails		This method is used to fetch the credit-policy based on the product group id and the business-unit via PMU adapter. Following steps are performed  <ol> <li>Fetch the CreditPolicyTemplateId based on the productGroupId and businessUnit</li> <li>Fetch the creditPolicyTemplateDTO based on the creditPolicyTemplateId</li> </ol> @param productGroupId @return @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateOffersForParties		This service will validate if all the applicants applying for one or more offers are eligible for setting up accounts using these products. Multiple applications contain different kinds of offers. Following validations are performed:  1. Validate the offer and party based on rule setup in rule engine. These are done at various restriction levels such as Product Group, branch , etc.  2. Check for any applicant warning indicators present for account opening events. If present, such applicants should be restricted from being added to accounts.  3. Call to check if party is eligible for all offers applied.   4. Check for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>accounts already setup for the party for the same product against maximum number of accounts allowed for the product setup in PMU module. If the maximum limit is exceeded, an error is thrown and the given product cannot be setup for the applicant.</p> <p>@fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ApplicantsProductValidationDTO, applicantsProductValidationDTO, , Indicates the list of applicants to be evaluated for the products applied for</p> <p>@return ApplicantsRuleRestrictionResponse, , Response object containing the results for the applicants from rule engine @throws FatalException</p> <p>@exception &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#APPLICANT_PRODUCT_INVALID} : FC_OR_DE_0067 &lt;br&gt; This error occurs when:&lt;br&gt; The offers being selected by the applicant cannot be applied due to restrictions setup&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#MAX_ACCTS_EXCEEDED} : FC_OR_DE_0065 &lt;br&gt; This error occurs when:&lt;br&gt; In case of deposit products (CASA and TD), each applicant can apply for limited number of accounts under the product defined in preferences setup in PMU. If the applicant tries to apply for more than the max number of accounts, this error will be seen &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#MAX_ACCTS_REACHED_BRN} : FC_OR_DE_0068 &lt;br&gt; This error occurs when:&lt;br&gt; In case of deposit products (CASA and TD), each applicant can apply for limited number of accounts under the product defined in preferences setup in PMU. If the applicant tries to apply for more</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>than the max number of accounts, this error will be seen. This error message gives more details on which applicant has breached the limit and how many accounts have already been setup &lt;br&gt; Presently, the validation is performed for DD, OD at offer level instead of product level. &lt;br /&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_CODE_NOT_FOUND} : FC_OR_DE_0054 &lt;br&gt; This error occurs when:&lt;br&gt; For the product code passed, no product group details were found or product code is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#INVALID_PARTY_ID} : FC_OR_DE_0057 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the party identifier cannot be fetched or the party identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PRODUCT_GROUP_NOT_FOUND} : FC_OR_DE_0059 &lt;br&gt; This error occurs when:&lt;br&gt; The details for the product group identifier cannot be fetched or the product group identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#WARNING_INDICATOR_APPLICABLE} : FC_OR_DE_0084 &lt;br&gt; This error occurs when:&lt;br&gt; Warning indicators which prevent account opening are present for applicants applying to the products.&lt;br&gt; &lt;br&gt; {@link OriginationBusinessPolicyConstants#APPLICANT_PRODUCTS_BUSINESS_POLICY} : FC_OR_BP_002 &lt;br&gt; This error occurs when:&lt;br&gt; Applicants are not eligible for applying the selected product groups due to any of the exception stated above.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	validateOffersForParties		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	The method is used to fetch all applicable offers for the approved application post assessment, in case if the requested amount is not same as approved amount (In upsell / downsell scenario).   If upsell / downsell scenario, then eligible offers are fetch from PMU and also populate the selectedOffer set in response. Else only return only the selected offer choosen / configured. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,applicationId, ,This attribute denotes the credit card application id @fcb.param MI,String,productGroupSerialNumber, ,This attribute denotes the credit card product group serial number @return {@link OfferReferenceInquiryResponse} This class contains applicable offer reference DTOs @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle Reference Id is used as an arrangement number for this pricing plan.  This reference Id is generated using the account module and is generated only once  for the given submission.  2. The Pricing Plan is created using this bundle reference Id.   3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.  4. Pricing Module is called to get pricing information for this Bundle Subscription Event.  5. If bundle subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.  6. Else Fees can be negotiated, waived or deferred for this bundle reference Id.   The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO,



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.  All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	This method validates the offer expiry. It performs the following operations  <ol> <li>Fetch the offer based on the offer-id via PMU adapter</li> <li>Check the offer start date, offer expiry date and the offer entity status</li> </ol> @param sessionContext @param valueDate @param offerId @param pgLinkageType @param validationErrorList @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails	This service will inquire the offer details for the offer identifier passed @param sessionContext @param productClass @param offerId @return OfferInquiryServiceResponse @throws FatalException	This service inquires the product Code for the offer identifier passed. Based on the simulation type, the inquiry to fetch casa, term deposit or loan offers is made to PMU. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SimulationType, simulationType, , Indicates if the simulation is of type loans, casa or term deposits. @fcb.param MI,String offerId, , Indicates the offer identifier for which product code is being inquired. @return String, , Indicates the product code for the offer Id. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCASA OfferInquiryResponse	This method would return the offer Inquiry response	This method validates the offer expiry. It performs the following operations  <ol> <li>Fetch the offer based on the offer-id via PMU adapter</li> <li>Check the offer start date, offer expiry date and the offer entity status</li> </ol> @param sessionContext @param valueDate @param offerId @param pgLinkageType @param validationErrorList @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCASA OfferInquiryResponse	This method would return the offer Inquiry response	This method performs the following operations:  <ol> <li>Fetch CASA offer back-dated period</li> <li>Fetch TD offer back-dated period</li> <li>Calculate the number of days between the value date and posting date</li> <li>If the number of days exceeds the allowed period, then throw error</li> </ol> @param sessionContext @param valueDate @param offerId @param productClass @param validationErrorList @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchCASA OfferInquiryResponse	This method would return the offer Inquiry response	This method will fetch the backdated account opening period allowed for the offer and validate with the value date and the posting date to check if the number of days allowed is breached @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchAllProductGroups	This method will return all the product groups @param sessionContext	
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchTDOfferInquiryResponse		This method validates the offer expiry. It performs the following operations  <ol> <li>Fetch the offer based on the offer-id via PMU adapter</li> <li>Check the offer start date, offer expiry date and the offer entity status</li> </ol> @param sessionContext @param valueDate @param offerId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param pgLinkageType @param validationErrorList @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchTDOfferInquiryResponse		This method performs the following operations: <ol> <li>Fetch CASA offer back-dated period</li> <li>Fetch TD offer back-dated period</li> <li>Calculate the number of days between the value date and posting date</li> <li>If the number of days exceeds the allowed period, then throw error</li> </ol> @param sessionContext @param valueDate @param offerId @param productClass @param validationErrorList @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchTDOfferInquiryResponse		This method will fetch the backdated account opening period allowed for the offer and validate with the value date and the posting date to check if the number of days allowed is breached @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchLoanReferences		@param sessionContext @param bundleOfferReferenceDTO @throws FatalException
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferValidity		@param sessionContext @param valueDate @param offerId @param termDeposits @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchDDAOfferDetails	@param offerCode @return	<p>This method validates that for the party passed and each chosen offer, the number of accounts to be applied plus existing accounts does not exceed the predefined maximum number of accounts per customer for a particular offer.&lt;br /&gt; A query is used to check the total number of accounts that are already opened for the applicant for a given offer whereby the relationship of the applicant to the account is either SOW, JAF or JAO.&lt;br /&gt; Also, accounts that are marked as 'CLOSED' are ignored in the count. &lt;br /&gt; Finally, a check for the total count is checked against the max limit set for each offer in PMU and if it exceeds, an error is thrown and the user is not allowed to setup the account. &lt;br /&gt; Originally, the validation was performed at product level instead of offer level. Additionally, the validation was performed for DDA, OD, TD, LOANS instead of DD and TD. @param sessionContext @param partyId @param productCode @param currencyCode @param accountSerialNumber @param productGroupLinkageType @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchActive Bundle		<p>This API Fetches Active Bundles for the Bundled flow from the PMU module using service <a href="#">{@link OfferInquiryApplicationService#fetchActiveBundle(SessionContext)}</a>. This method internally performs the following operations</p> <ul style="list-style-type: none"> <li>It fetches all the Bundles for the current financial year from the session.</li> <li>It populates all bundles into <a href="#">{@link BundleOfferDTO}</a> and send it back through response</li> </ul> <p>there are no Prerequisites for this API. <code>@param sessionContext MI,SessionContext,sessionContext</code>, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. Session Context must contain the session context information such as user id, branch, branch code, channel etc. <code>@return BundleOfferResponse</code> It contains an array of <a href="#">{@link BundleOfferDTO}</a> and Transaction Status. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li><b>Transaction Success</b> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</a></li> </ul> </li> <li><b>Transaction Failure</b> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</a></li> </ul> </li> </ul> <p><code>@throws</code> <a href="#">{@link com.ofss.fc.infra.exception.FatalException}</a> Please refer to the documentation above for <code>&lt;i&gt;Transaction Failure&lt;/i&gt;</code> section</p>
IOrigination PMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchActive Bundle		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		The method is used to fetch all applicable offers for the approved application post assessment, in case if the requested amount is not same as approved amount (In upsell / downsell scenario).   If upsell / downsell scenario, then eligible offers are fetch from PMU and also populate the selectedOffer set in response. Else only return only the selected offer choosen / configured. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,applicationId, ,This attribute denotes the credit card application id @fcb.param MI,String,productGroupSerialNumber, ,This attribute denotes the credit card product group serial number @return {@link OfferReferenceInquiryResponse} This class contains applicable offer reference DTOs @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		This service is used for creating pricing plan for bundle subscription fees.  Either existing account linked to bundle or new offer can be nominated for bundle fees  The following activities will be carried out in sequence.  1. The Bundle

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Reference Id is used as an arrangement number for this pricing plan.&lt;br&gt; This reference Id is generated using the account module and is generated only once &lt;br&gt; for the given submission.&lt;br&gt; 2. The Pricing Plan is created using this bundle reference Id. &lt;br&gt; 3. A new pricing ticket number is generated and returned by this service. If existing offer is nominated for bundle fees then the pricing ticket is stored against the product group linkage of that offer.&lt;br&gt; 4. Pricing Module is called to get pricing information for this Bundle Subscription Event.&lt;br&gt; 5. If bundle subscription fee treatment is defined as 'Capitalize' for the product group, intimate pricing will be called to mark the chargeActionType as Capitalize.&lt;br&gt; 6. Else Fees can be negotiated, waived or deferred for this bundle reference Id. &lt;br&gt; The pricing ticket number stored for bundle fees is used for tracking such fees collected against this arrangement number.&lt;br&gt; @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BundleDetailsDTO,bundleDetailsDTO, ,Information related to Bundle subscribed during submission creation @fcb.param MI,ProductGroupLinkageDTO,productGroupLinkage DTO, ,Product Group information for the group nominated for bundle subscription fees. @return {@link PricingResponse} This class holds ticket number for bundle fees which is stored.&lt;br&gt; All inquiries for this bundle reference Id are made using this ticket number. @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		This method validates the offer expiry. It performs the following operations  <ol> <li>Fetch the offer based on the offer-id via PMU adapter</li> <li>Check the offer start date, offer expiry date and the offer entity status</li> </ol> @param sessionContext @param valueDate @param offerId @param pgLinkageType @param validationErrorList @throws FatalException
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		
IOriginationPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationPMUAdapter	fetchOfferDetails		This service inquires the product Code for the offer identifier passed. Based on the simulation type, the inquiry to fetch casa, term deposit or loan offers is made to PMU. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, SimulationType, simulationType, , Indicates if the simulation is of type loans, casa or term deposits. @fcb.param MI,String offerId, , Indicates the offer identifier for which product code is being inquired. @return String, , Indicates the product code for the offer Id. @throws FatalException
IPricingOPSApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingOPSApplicationAdapter	startOrRestartProcess	This method is used to initiate the batch from pricing poller. @param sessionContext @param categoryCode @param type @param group @param groupCode	This function listens to the Price change and find out the accounts to be re-priced & saves the re-pricing data. @param sessionContext @throws FatalException
IPricingOPSApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingOPSApplicationAdapter	startOrRestartProcess	This method is used to initiate the batch from pricing poller. @param sessionContext @param categoryCode @param type @param group @param groupCode	Starts or restarts the Reprice Category @param sessionContext @param eodCommandType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingOPSApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingOPSApplicationAdapter	startOrRestartProcess	This method is used to initiate the batch from pricing poller. @param sessionContext @param categoryCode @param type @param group @param groupCode @param group @param groupCode	This method is used to initiate the batch from pricing poller. @param sessionContext @param categoryCode @param type @param group @param groupCode @throws FatalException
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductCode		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductCode		this is the main validate method @param productEventLinkage
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		This method fetches accounting entry template details on product code,status and/or event code. The method fetches accounting entry templates based on product code,status and/or event code. For each accounting entry template the credit and debit account role attributes are retrieved and also the credit and debit amount tag attributes.  If the account role retrieved has derivation type as MODULE_SUPPLIED, then corresponding ledger code for the product, account role and status is returned as null from the account role to head mapping   If the account role retrieved has derivation type as DERIVED, then corresponding ledger code for the product, account role and status is fetched from the account role to head mapping.  If the accounting entry templates are not found then null object is returned. @param productCode String @param eventCode String @param status String @return returnTemplateDetails IAccountingEntryTemplateDetails[]
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		This method fetches all the amount tags for which the corresponding account role is module supplied. The method queries the product group accounting entry templates and returns those amount tags for which the corresponding account role is MODULE_SUPPLIED. @param productGroupCode String @param eventCode String @return amountTags IAmountTagInformation[]
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		This method performs merge operation. @param productGroupRoleHead
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		This method performs merge operation. @param productGroupEventLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductGroup		@param productGroupEventLinkage @return
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		This method returns the ledger code for the given product code.  If the account role is mapped directly to the account head then the account head is returned as the ledger code.  If the account role is mapped to the rule code, then the rule code is used to fetch ledger code based on the rule code and map of fact id and fact value pair.  @param productCode @param accountRole @param factsMap @return
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchProductClass		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchAllowedProductClass		This method fetches the list of account roles based on the product group @param productGroup String[] @return accountRoles String[]
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	fetchAllowedProductClass		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		This method performs add operation. @param productGroupRoleHead
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		This method performs update operation. @param productGroupRoleHead
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		This method performs merge operation. @param productGroupRoleHead

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroup		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCodeAtAllLevels		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCodeAtAllLevels		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		
IProductManufactureApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IProductManufactureApplicationAdapter	validateProductGroupCode		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmount	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param ppfDate @param localCurrencyCode @return	This function will call DB package to calculate penalty amount for PPF @param applicationContext @param account @param effectiveDate @param penaltyMethodType @param principalPPFAmount @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmount	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param ppfDate @param localCurrencyCode @return	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param penaltyMethod @param ppfDate @return penaltyAmount @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmount	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param ppfDate @param localCurrencyCode @return	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param penaltyMethod @param ppfDate @return penaltyAmount @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmount	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param ppfDate @param localCurrencyCode @return	This method calculates the penalty amount for partial pay-off based on the amount passed less the total arrears. @param context @param accountId @param ppfAmount @param penaltyMethod @param ppfDate @return Money @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmount	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param ppfDate @param localCurrencyCode @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmountOnSimulation	This method calculates the penalty amount for partial pay-off based on the amount passed to it. This method to be used in simulation mode only. @param context @param accountId @param ppfAmount @param ppfDate @return ppfPenalty Money	This function will call DB package to calculate penalty amount for PPF @param applicationContext @param account @param effectiveDate @param penaltyMethodType @param principalPPFAmount @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPPFPenaltyAmountOnSimulation	This method calculates the penalty amount for partial pay-off based on the amount passed to it. This method to be used in simulation mode only. @param context @param accountId @param ppfAmount @param ppfDate @return ppfPenalty Money	This method calculates the penalty amount for partial pay-off based on the amount passed to it. @param context @param accountId @param ppfAmount @param penaltyMethod @param ppfDate @return penaltyAmount @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getServiceChargeDetails	This method gets the deduction details from pricing It passes loan account related facts to pricing module whichReturns a list of deduction details. @param applicationContext @param accountDetailsDTO @return List<ServiceChargeDetailsData> @throws FatalException	This method gets the deduction details from pricing @param context @param accountId @param deductionEvent @param deductionDetails @return List<ServiceChargeDetailsDTO> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getServiceChargeDetails	This method gets the deduction details from pricing It passes loan account related facts to pricing module whichReturns a list of deduction details. @param applicationContext @param accountDetailsDTO @return List<ServiceChargeDetailsData> @throws FatalException	This method gets the deduction details from pricing It passes loan account related facts to pricing module whichReturns a list of deduction details. @param applicationContext @param accountDetailsDTO @return List<ServiceChargeDetailsData> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getServiceChargeDetails	This method gets the deduction details from pricing It passes loan account related facts to pricing module whichReturns a list of deduction details. @param applicationContext @param accountDetailsDTO @return List<ServiceChargeDetailsData> @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	<pre>This method is used to restructure the loan account based on the new schedule code &lt;br&gt; It has the following input parameters:- &lt;br&gt; 1)LoanOfferSwapDetailsDTO The following steps are carried out in this method:- &lt;br&gt; 1)The doRestructure() method of LoanRepaymentScheduleApplicationService is called with the schedule code, account number &amp; reschedule date i.e., process date. &lt;br&gt; </pre> @param loanOfferSwapDetailsDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	This method performs the operation of disbursing the parent Loan Account with the new sanctioned amount @param applicationContext @param parentAccountId @param totalPaidOffAmount @param processDate @param feeCollectionDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	This method saves the interest, margin and price offerings. @param context @param accountId @param effectiveDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	<pre>This method is used to restructure the loan account based on the new schedule code &lt;br&gt; It has the following input parameters:- &lt;br&gt; 1)LoanOfferSwapDetailsDTO The following steps are carried out in this method:- &lt;br&gt; 1)The doRestructure() method of LoanRepaymentScheduleApplicationService is called with the schedule code, account number &amp; reschedule date i.e., process date. &lt;br&gt; </pre> @param loanOfferSwapDetailsDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferings	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	repriceOfferings	<p>Re-prices the regular and penalty base interest index offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing interest index offerings, re-evaluating the regular and penalty interest policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the regular and penalty margin index offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing margin offerings, re-evaluating the regular and penalty margin policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. &lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts.</p> <p>&lt;/p&gt; @param sessionContext the session context @param pricingContextDTO the pricing context dto @param arrangementNumber the arrangement number @param repriceDate the date on which to reprice @throws FatalException if the input is invalid, or there is a problem while repricing</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingPlan	<p>@param applicationContext</p> <p>@param loanAccountOpeningDTO</p> <p>@param partyId @throws FatalException</p>	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. It works as follows:</p> <ul style="list-style-type: none"> <li>Create AdapterFactory and get an instance of LoanPricingApplicationAdapter</li> <li>Call {@link ILoanPricingApplicationAdapter#createPricingPlanAndEndDialogue(priceTicketNumber: String, accountNumber: String)} to create the pricing plan</li> <li>Creating PricingPlan Works as follows: <ul style="list-style-type: none"> <li>If pricing ticket is not null <ul style="list-style-type: none"> <li>Indicating that the rate is already negotiated with the pricing module and same is required to be applied.</li> <li>Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</li> <li>pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null</li> <li>set other attributes in the input.</li> </ul> </li> <li>If pricing ticket is null <ul style="list-style-type: none"> <li>Indicating that there is no rate communication happened between pricing and the calling module.</li> <li>Default rate is required to be applied.</li> <li>Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()}</li> <li>Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</li> </ul> </li> </ul> </li> <li>to apply the variance and creating pricing plan</li> </ul> <p>@param context</p> <p>@param loanAccountOpeningDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingPlan	@param applicationContext @param loanAccountOpeningDTO @param partyId @throws FatalException	@param applicationContext @param loanAccountOpeningDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre>This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price definitions are evaluated and the fee amount is returned back. &#064;param applicationContext &#064;param accountId &#064;param eventId &#064;param transactionAmount @return	This method constructs, {@link FeeCollectionDTO} which holds the attributes pertaining to fee collection during various Loan OLTP transactions. @param applicationContext @param accountId @param eventId @param transactionAmount @return feeCollectionDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre>This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price definitions are evaluated and the fee amount is returned back. &#064;param applicationContext &#064;param accountId &#064;param eventId &#064;param transactionAmount @return	This method is used to process late charges if any. It is called only for account opened in offer which has Basis for penalty as Penalty fee. While Raising arrear, calculate and persist tolerance amount in loan arrear. If sum of all unpaid arrear is greater than sum of their corresponding tolerance amount, then call pricing to charge penalty fee. @param context @param accountId @param actionDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method is responsible for returning fees for an event @param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException </pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method is responsible for returning fees for an event @param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method will doAccountClosers @param applicationContext @param account @param eventId @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method gets the deduction details from pricing. It passes loan account related facts to pricing module which returns a list of deduction details. @param applicationContext @param accountDetailsDTO @return List&lt;ServiceChargeDetailsData&gt; @throws FatalException </pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method calls the pricing module service to get the fee amount on basis of the eventCode passed. @param context @param accountId @param effectiveDate @param loanInterestType @param eventCode @return feeAmount </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method calls the pricing module service to get the fee amount on basis of the eventCode passed. @param context @param accountId @param effectiveDate @param loanInterestType @param eventCode @return feeAmount </pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	The method is responsible to fees for a given account recommendation and a given event id. This would be useful in case of new account configuration or some variation events wherein the account configuration for event fees are to be fetched do not yet have a pricing plan.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<p>The method is responsible to fees for a given account recommendation and a given event id. This would be useful in case of new account configuration or some variation events wherein the account configuration for event fees are to be fetched do not yet have a pricing plan.</p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEvent	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<pre> This method constructs, {@link FeeCollectionDTO} which holds the attributes pertaining to fee collection during various Loan OLTP transactions. @param applicationContext @param accountId @param eventId @param transactionAmount @return feeCollectionDTO @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<p>This method is used to process late charges if any. It is called only for account opened in offer which has Basis for penalty as Penalty fee. While Raising arrear, calculate and persist tolerance amount in loan arrear. If sum of all unpaid arrear is greater than sum of their corresponding tolerance amount, then call pricing to charge penalty fee. @param context @param accountId @param actionDate</p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	<p>This method is responsible for returning fees for an event @param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre>&lt;pre&gt;This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. ;param applicationContext ;param accountId ;param eventId ;param transactionAmount @return</pre>	<pre>This method will doAccountCloser @param applicationContext @param account @param eventId @throws FatalException</pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre>&lt;pre&gt;This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. ;param applicationContext ;param accountId ;param eventId ;param transactionAmount @return</pre>	<pre>This method calls the pricing module service to get the fee amount on basis of the eventCode passed. @param context @param accountId @param effectiveDate @param loanInterestType @param eventCode @return feeAmount</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	The method is responsible to fees for a given account recommendation and a given event id. This would be useful in case of new account configuration or some variation events wherein the account configuration for event fees are to be fetched do not yet have a pricing plan.
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id Based on the above information price defintions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	The method is responsible to fees for a given account recommendation and a given event id. This would be useful in case of new account configuration or some variation events wherein the account configuration for event fees are to be fetched do not yet have a pricing plan.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventSimulation	<pre> This method is responsible for fetching fees for an event. Fees are defined in pricing module for a particular event. The information sent to pricing to fetch fees are 1) Loan Facts 2) Account Id 3) Transaction Amount 4) Event Id. Based on the above information price definitions are evaluated and the fee amount is returned back. @param applicationContext @param accountId @param eventId @param transactionAmount @return </pre>	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceOnEventStrictlyAtReco	<pre> @param context @param accountId @param eventId @param transactionAmount @return @throws FatalException </pre>	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTOToPriceFeatureDTO	<pre> @param wrappedPriceFeatureDTO @return </pre>	<p>This method will be responsible for processing the reverse transaction. This method will first check if the original transaction was back-dated or not, if it was back-dated then, delete the back-dated instruction and update the day balance accordingly. And if the original transaction was a current dated transaction then, first validating if reversal is allowed of that transaction or not, if validation is successful then, posting the reverse transaction.</p> <pre> @param applicationContext @param accountId @param eventId @param transactionDetailsDTO @param originalTransactionReferenceNumber @param componentPreferenceDTO @param feeCollectionDTO </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTOToPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	This method is used to process late charges if any. It is called only for account opened in offer which has Basis for penalty as Penalty fee. While Raising arrear, calculate and persist tolerance amount in loan arrear. If sum of all unpaid arrear is greater than sum of their corresponding tolerance amount, then call pricing to charge penalty fee. @param context @param accountId @param actionDate
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTOToPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	This method is used for processing the online Fees: Fees are defined in pricing module for list of module events Typical interaction with pricing module will involve a)Fetching of Fees for an event b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete @param applicationContext @param accountId @param eventId @param wrappedPriceFeatureDTO @param deductionModeType @param transactionAmount @param pricingTicketNumber @param priceFeatureDTO @param accountCurrency @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTOToPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	This method will convert from WrappedPriceFeatureDTO to priceFeatureDTO. @param wrappedPriceFeatureDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTOToPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	This method is responsible for deriving Periodic Fee Amount. @param applicationContext @param accountId @param eventDate @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	@param wrappedPriceFeatureDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromWrappedPriceFeatureDTO	@param wrappedPriceFeatureDTO @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	notifyPricing	@param accountId @param pricingTicketNumber @param transactionReferenceNumber @param priceDefinitionId @param feeAmount @param eventId @param feeChargeDate @param feeChargeDetailId TODO @throws FatalException	This method is used to process late charges if any. It is called only for account opened in offer which has Basis for penalty as Penalty fee. While Raising arrear, calculate and persist tolerance amount in loan arrear. If sum of all unpaid arrear is greater than sum of their corresponding tolerance amount, then call pricing to charge penalty fee. @param context @param accountId @param actionDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	notifyPricing	@param accountId @param pricingTicketNumber @param transactionReferenceNumber @param priceDefinitionId @param feeAmount @param eventId @param feeChargeDate @param feeChargeDetailId TODO @throws FatalException	<pre>This method is responsible for management of interactions with pricing module for prices and levying the fees on the account Fees are defined in pricing module for different period Typical interaction with pricing module will involve a)Fetching of Fees for a period b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete The inputs to the method are :- a)ApplicationContext b)accountId c)eventId d)List<&lt;PriceFeatureDTO>&gt; e)DeductionModeType (Bill , Deduct , Capitalize) &#064;param applicationContext &#064;param accountId &#064;param deductionModeType &#064;param wrappedPriceFeatureDTO @param deductionModeType
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	notifyPricing	@param accountId @param pricingTicketNumber @param transactionReferenceNumber @param priceDefinitionId @param feeAmount @param eventId @param feeChargeDate @param feeChargeDetailId TODO @throws FatalException	Periodic fees cannot be levied when account is in hardship with given hardship reliefs as Rollover or Freeze Repayment. The periodic fees during the relief should be marked to levy post hardship relief. This method will mark periodic fees Defer if they are of Bill nature else will mark them as capitalized. @param applicationContext @param accountId @param computationDate @param deductionModeType @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	notifyPricing	@param accountId @param pricingTicketNumber @param transactionReferenceNumber @param priceDefinitionId @param feeAmount @param eventId @param feeChargeDate @param feeChargeDetailId TODO @throws FatalException	@param applicationContext @param accountId @param deductionModeType @param featureDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	notifyPricing	@param accountId @param pricingTicketNumber @param transactionReferenceNumber @param priceDefinitionId @param feeAmount @param eventId @param feeChargeDate @param feeChargeDetailId TODO @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	<p>This method will do processing when bundle requests for bundle subscription fee charging from the first disbursed loan account under the nominated offer.  This method will do the following <ul><li>raise arrear for the fee to be charged.</li> <li>log interest computation action.</li> <li>update utilization for the account charged.</li> </ul>  Accounting will not be raised here, both Dr PD_ID_RECEIVABLE, Cr BN_INTERMEDIARY_BUNDLE_GL and Dr BN_INTERMEDIARY_BUNDLE_GL, Cr Fee_Income will be raised by BUNDLE module. </p> @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param additionalFee @param baseNetFeesAmount @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deffered fees for an account @param sessionContext @param accountId @throws FatalException	This method is used to charge and capitalize the total fees accrued in the moratorium period on moratorium stage end date. @param accountId @param currencyCode @param compoundDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deffered fees for an account @param sessionContext @param accountId @throws FatalException	<pre>This method is responsible for management of interactions with pricing module for prices and levying the fees on the account Fees are defined in pricing module for list of module events Typical interaction with pricing module will involve a)Fetching of Fees for an event b)Levying of fees on the account c)Intimating pricing with information of fee levied on the acocunt d)Notifying pricing that a fee transaction is complete The inputs to the method are :- a)ApplicationContext b)accountId c)eventId d)List<PriceFeatureDTO>; e)DeductionModeType (Bill , Deduct , Capitalize) It is expected that fee taskflow embeded in screens will have set the values correctly in PriceFeatureDTO in relation to the negotiated amount and waived amount If(eventId == LOAN_DISBUSREMENT) then invoke manageDisbusrementFees() else Iterate through the PriceFeatureDTO do the following :- 1)Get the fee amount to be leveied , fee amount to waived in a local variable. 2)If feeAmountToBeLevied > ZERO   If DeductionModeType == BILL_FEES then invoke arrear.raise and pass the required details invoke intimatePricing and pass the required details If DeductionModeType == DEDUCT_FEES then   invoke arrear.raiseAnPay and pass the required details invoke intimatePricing and pass the required details If DeductionModeType == CAPITALIZE_FEES then invoke arrear.capitalize and pass the required details invoke intimatePricing and pass the required details 3)If feeAmountToBeWaived > ZERO   invoke Arrear.record and pass the required details &#064;param applicationContext &#064;param accountId &#064;param wrappedPriceFeatureDTO @param deductionModeType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deferred fees for an account @param sessionContext @param accountId @throws FatalException	<p>The method is responsible to levy deferred fees during hardship relief period. This method will levy following deferred fees <ul> <li>Loan Periodic fees</li> <li>Penalty Fees</li> <li>Facility Fees</li> <li>Bundle maintenance fees</li> </ul>  @param applicationContext @param accountId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deferred fees for an account @param sessionContext @param accountId @throws FatalException	@param applicationContext @param eventId @param accountId @param disbursementAmount @param loanAccount @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deferred fees for an account @param sessionContext @param accountId @throws FatalException	This method is used to get the details of the deferred fees @throws FatalException @ accountId
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchDeferredFeesForAccountId	This method is used to fetch the fetch the deferred fees for an account @param sessionContext @param accountId @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method will merge the rates during processing of rollover. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method will merge the rates during rollover simulation. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method will merge the rates during processing of rollover. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> &lt;li&gt;FacilityId&lt;/li&gt; &lt;li&gt;OutstandingAmount&lt;/li&gt; &lt;li&gt;CurrentFixedRate&lt;/li&gt; &lt;li&gt;QuoteExpiryDate&lt;/li&gt; &lt;/pre&gt; @param context @param accountId @return</pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used to get reprice frequency from pricing module. @param applicationContext @param arrangementNumber @param installmentCalculationMethod @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method returns the interest rates applicable on the current process date in form of {@link NetRateDTO}. @param applicationContext @param accountId @return netRateDTO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method returns the account information required by collections. This method will be called by party to fetch accounts information for sharing with collection. @param sessionContext @param accountId return AccountDetailsInquiryResponse
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used to log the non-financial statement for net rate change. @param sessionContext @param accountId @param valueDate @param eventId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used to get the margin for logging it in the statement. @param applicationContext @param accountId @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method will give Rate as on date. @param applicationContext @param accountId @param onDate @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the existing rates from the pricing module for stages other than PMI.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param stageNames @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the existing rates from the pricing module for PMI stage.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param stageNames @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param stageNames @return SortedMap<LoanInterestType, List<NetRateDTO>>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param offerId @param pricingTicketNumber @param productId @param fixedTermInMonths @param newApprovedAmount @param newLoanTerm @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the existing rates from the pricing module as on Back value date, from the rates history.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository as on back value date from history otherwise the supplied stageNames are used. </li> @param applicationContext @param accountId @param valueDate @param loanInterestType @param stageNames @return SortedMap<LoanInterestType, List<NetRateDTO>>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method returns the net rate for account for the given stage and date. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @param dateToFilterOn @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	<p>This method is responsible for fetching Rates using Loan Account Configuration details.   It uses Loan Account Opening details from Loan Account Configuration details for preparing facts. </p> @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return

<b>Adapter Name</b>	<b>Adapter Path</b>	<b>Adapter Method</b>	<b>Interface Method Javadoc</b>	<b>Implementation Method Javadoc</b>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used to fetch rate for a given stage and date combination. @param applicationContext @param arrangementNumber @param asOnDate @param installmentCalculationMethod @param pricingElementType @param dateToFilterOn @return NetRateDTO[]
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used to fetch the rates for the input date and the day before. @param sessionContext @param arrangementNumber @param asOnDate @return NetRateStatementInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method is used for current dated as well as backdated facility level amount or term change. @param arrangementNumber @param emptyForm @param repriceDate @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRates	This method performs the operation of fetching Rates from Pricing. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchScheduledRatesForStage	This method performs the operation of fetching Rates from Pricing with the changed Rates @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	This method performs the operation of fetching the existing rates from the pricing module with changed rates <ul style="list-style-type: none"> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param stageNames @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchScheduledRatesForStage	This method performs the operation of fetching Rates from Pricing with the changed Rates @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	<p>This method is responsible for providing the details required for Document Generation.   It uses Loan Schedule Configuration and Loan Account Configuration to derive the Rate details and Repayment details for every rate details. </p> @param sessionContext @param arrangementId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	@param applicationContext @param accountId @param eventToComputeInterest @param loanAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	This method performs the operation of fetching the simulated Rates from Pricing of PMI stage. The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	@param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @param isPostMaturity @param isMergingRequired @return SortedMap<LoanInterestType, List<NetRateDTO>>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param loanAccountOpeningDTO @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	<p>This method is responsible for fetching Rates using Loan Account Configuration details.   It uses Loan Account Opening details from Loan Account Configuration details for preparing facts.</p> @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRates	This method performs the operation of fetching the simulated Rates from Pricing. @param applicationContext @param loanPricingDetailsInputDTO @param isMergingRequired @return NetRateDTO[]	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method performs the operation of processing the loan rollover instruction. <li>The stage as on the rollover date is obtained.</li> <li>Stage would be split on either repayment review Date or Last Charging Date.</li> <li> nextChargingDate is obtained which is after the given start date after which the given type of schedule event will occur and then Rollover Business Policy is applied.</li> <li>There are cases on which the loanStage are either elapsed or not.</li> <li>If RolloverType equals MOVE_TO_FIXED then four Cases are handled.</li>   Case 1: Stage s1 is partially in newFixedTerm i.e. s1 starts before newFixedTerm & ends in newFixedTerm  case:2 Stage s1 is partially in newFixedTerm i.e. s1 starts in newFixedTerm & ends after newFixedTerm  case3: Stage s1 is partially in newFixedTerm i.e. s1 starts before newFixedTerm & ends after newFixedTerm  case:4 Stage s1 is fully in newFixedTerm i.e. s1 starts in newFixedTerm & ends in newFixedTerm  <li>If fixedTermRange contains stageStartDate or fixedTermRange contains stageEndDate or fixedTermStartDate is after Or Equal to stageStartDate and fixedTermEndDate is Before Or Equal to stageEndDate</li> <li> AllowInterestRate is set to true.</li> <li>AllowMarginRate is set to true.RetainUDM is set to false. stageStartDate is in newFixedTermRange and processing is done.</li> <li>If RolloverType equals MOVE_TO_VARIABLE then four cases are handled.</li>   Case1: Stage s1 is partially in newVariableTerm i.e. s1 starts before newVariableTerm & ends in newVariableTerm  case2 : Stage s1 is partially in newVariableTerm i.e. s1 starts in newVariableTerm & ends after newVariableTerm  case 3: Stage s1 is partially in newVariableTerm i.e. s1 starts before

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>newVariableTerm &amp; ends after newVariableTerm&lt;br&gt;           case 4: Stage s1 is fully in newVariableTerm i.e. s1 starts in newVariableTerm &amp; ends in newVariableTerm&lt;br&gt;           &lt;li&gt;If variableTermRange contains stageStartDate or variableTermRange contains stageEndDate or variableTermStartDate is after Or Equal stageStartDate and variableTermEndDate is before or equal stageEndDate AllowInterestRate is set to true AllowMarginRate is set to false RetainUDM is set true stageStartDate is checked whether it is in the newvariableTermRange Schedule fees is called and the service is called.&lt;/li&gt;           &lt;li&gt;             LoanInterestComputationService.logComputationActionOnLoanReschedule is called to recompute the interest for the schedule.&lt;/li&gt;</p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	<p>This method is used to persist the rates in the pricing module            @param applicationContext            @param loanPricingDetailsInputDTO            @return NetRateDTO[]</p>	<p>This method is used to persist the Rates in Pricing module. &lt;br&gt;           &lt;ul&gt;             &lt;li&gt;If the input fixedTermInMonths is null then the existing value is retrieved via LoanAccountAdditionalDetailsRepository. Otherwise the supplied value is used.&lt;/li&gt;             &lt;li&gt;If the input repaymentStages is null then the existing value is retrieved via LoanRepaymentScheduleRepository.&lt;/li&gt;             &lt;li&gt;persistRates is called through loanPricingApplicationAdapter where the data is persisted in the database&lt;/li&gt;             &lt;li&gt;&lt;/li&gt;           &lt;/ul&gt;           @param applicationContext @param accountId            @param effectiveDate @param loanInterestType            @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount            @return SortedMap&lt;LoanInterestType, List&lt;NetRateDTO&gt;&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method is used to persist the PMI Rates in Pricing module.   <ul> <li>If the input fixedTermInMonths is null then the existing value is retrieved via LoanAccountAdditionalDetailsRepository. Otherwise the supplied value is used.</li> <li>If the input repaymentStages is null then the existing value is retrieved via LoanRepaymentScheduleRepository.</li> <li>persistRates is called through loanPricingApplicationAdapter where the data is persisted in the database</li> <li></li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method is used to persist the Rates in Pricing module.   <ul> <li>If the input fixedTermInMonths is null then the existing value is retrieved via LoanAccountAdditionalDetailsRepository. Otherwise the supplied value is used.</li> <li>If the input repaymentStages is null then the existing value is retrieved via LoanRepaymentScheduleRepository.</li> <li>persistRates is called through loanPricingApplicationAdapter where the data is persisted in the database</li> <li></li> </ul> @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @param newDisbursedAmount @param isPostMaturity @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method persists the rates in the pricing module
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method will call pricing module to persist the NetRate. @param applicationContext @param accountId @param interestRates @param effectiveDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt;Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistRates	This method is used to persist the rates in the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return NetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	<pre>This method is responsible for management of interactions with pricing module for prices and levying the fees on the account Fees are defined in pricing module for different period Typical interaction with pricing module will involve a)Fetching of Fees for a period b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete The inputs to the method are :- a)ApplicationContext b)accountId c)eventId d)List<PriceFeatureDTO>; e)DeductionModeType (Bill , Deduct , Capitalize) &#064;param applicationContext &#064;param accountId &#064;param deductionModeType &#064;param wrappedPriceFeatureDTO @param deductionModeType
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	Fetch Periodic fees for MOR,Hardship stage etc. @param sessionContext @param context @param accountId @param evaluationDate @param currencyCodes @param computationDate @param amount @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	Periodic fees cannot be levied when account is in hardship with given hardship reliefs as Rollover or Freeze Repayment. The periodic fees during the relief should be marked to levy post hardshhip relief. This method will mark periodic fees Defer if they are of Bill nature else will mark them as capitalized. @param applicationContext @param accountId @param computationDate @param deductionModeType @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method performs the getDummyPeriodicPricesSimulation operation @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	This method fetches the periodic prices with capitalized prices.
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPrices	This method is used to getPeriodicPrices maintained at the pricing module @param applicationContext @param loanPricingDetailsInputDTO @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateRefundFeeDetails	@param sessionContext @param reversalDetailsDTOs @throws FatalException	This method is used to update the refund details @param sessionContext @param reversalDetailsDTOs @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateRefundFeeDetails	@param sessionContext @param reversalDetailsDTOs @throws FatalException	@param accountId @param backValueDate
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	<pre>This method is responsible for reversal of disbursement. Reversal of disbursement will involve deleting all data inserted during the disbursement process. &#064;param applicationContext &#064;param accountId &#064;param originalTransactionReferenceNumber @param componentPreferenceId @param isSettlementPayoutReversal @return void
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	This method reverses transactions details for a given external reference number. @param externalReferenceNumber @return flag denoting whether fee is reversed or not.
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	This method is called to intimate pricing of fee reversal. @param accountId @param priceFeatureDTO @param eventId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransaction	@param pricingContext @param ticketNumber @param eventCode @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransactionByReferenceNumber	@param sessionContext @param accountId @param externalReferenceNumber @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalTransactionByReferenceNumber	@param sessionContext @param accountId @param externalReferenceNumber @return	<pre>This method is responsible for reversal of disbursement Reversal of disbursement will involve deleting all data inserted during the disbursement process &#064;param applicationContext &#064;param accountId &#064;param originalTransactionReferenceNumber @param componentPreferenceId @param isSettlementPayoutReversal @return void
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	This method is responsible for deriving Fees for Early and Final PayOff. @param applicationContext @param accountId @param efsPenaltyBase @param efsSimulationDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	This method calculates the penalty amount for full pay-off. @param context @param accountId @param eventId @param valueDate @param efsPenaltyBase @return penaltyAmount @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	This method is responsible for deriving Fees for Early and Final PayOff. @param applicationContext @param accountId @param efsPenaltyBase @param efsSimulationDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	<p>This method is responsible for performing following tasks:-   <ol> <li>Check if Closure Request has already been raised or not.</li> <li>If Request ID is found, it's passed for further validations. If not, other validations are being performed.</li> <li>Calculate Final Settlement Amount</li> <li>Send Final Settlement Amount as Response.</li> </ol></p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	<p>This method is responsible for deriving Fees for "Early & Final Settlement".   If the Event Based Fees are not being provided from Screen or from any External Interface (which uses this service), the fee can be derived internally using this method. </p> @param applicationContext @param accountId @param currencyCode @param valueDate @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	@param accountId @param valueDate @param loanEventCodes @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePriceOnEvent	@param context @param accountId @param eventId @param eventDate @param transactionAmount @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method is responsible for deriving pending periodic fees. @param context @param accountId @param valueDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method does levying of periodic fees. @param applicationContext @param accountId @param valueDate @param currencyCode
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method will fetch sum total of Pending periodic Fee for account closure. @param context @param accountId @param valueDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method will fetch Pending periodic Fee for account closure. @param applicationContext @param accountId @param eventDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method returns prorated fees at maturity date by calling the pricing services through an adapter. @param context @param accountId @param penaltyMethod @param valueDate @return Money throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method is responsible for deriving Pending Periodic Fees. @param applicationContext @param accountId @param currencyCode @param efsSimulationDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	<p>This method is responsible for closing Loan Account.   During this process, it performs the following steps:-   <ol> <li>Fetches Loan Account Closure Request details, if raised already.</li> <li>Following validations are performed:-   <ol> <li>Checks whether "Early & Final Settlement" is allowed for offer, to which the Loan Account belongs to.</li> <li>Checks whether Loan Account is in Lock-In period.</li> <li>Checks whether Interest has been charged or not.   (i.e. checks if Last Interest Charging Date > Current Date)</li> <li>Checks for Uncleared Amount.</li> <li>IF Request has been raised THEN   <ul> <li>Check if Retention Strategy is mentioned or not</li> <li>Check if Retention Strategy is marked as 'ProceedWithClosure' or not, if mentioned</li> <li>Check if Loan Account Status is 'MarkedForClosure' or not.</li> <li>Check if Loan Account has credit facility, User Status to Event Mapping.</li> </ul> ELSE IF Request has not been raised THEN   <ul> <li>Check if Loan Account Status is 'Regular' or not.</li> <li>Check if Loan Account has credit facility, User Status to Event Mapping.</li> </ul> </li> </ol> </li> <li>Validate Transaction Details.</li> <li>Inquire Full Pay Off Details.  This provides following:- <ul> <li>Total Due Amount</li> <li>EFS Penalty Amount</li> <li>Arrears</li> <li>Economic Cost</li> </ul> </li> <li>Inquire Loan Preferences and get Forfeit balance</li> <li>Based on Forfeit Balance and Economic Cost and Forfeit Balance, perform following Validations:-  <ul> <li>IF (Transaction Amount > Total Due) AND (Transaction Amount - Total Due) > Forfeit Balance THEN   Raise Error. This confirms that Customer should not

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>pay more than Total Due expected.&lt;/li&gt; &lt;li&gt;IF (Total Due - Transaction Amount) &gt; Forfeit Balance THEN</p> <p>&lt;br&gt; Raise Error. This confirms that Customer should pay Total Due expected.&lt;br&gt; Bank can afford the loss till the amount equivalent to forfeit amount, if Customer pays less than what is expected.&lt;/li&gt;</p> <p>&lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Process "Early &amp; Final Settlement" transaction.&lt;/li&gt; &lt;li&gt;Update Loan Account Preferences with Reason for Account Closure.&lt;/li&gt; &lt;li&gt;Delete all Future dated Instructions maintained for the Loan Account:-&lt;br&gt; &lt;ul&gt; They can be &lt;li&gt;Auto-Disbursement Instructions&lt;/li&gt; &lt;li&gt;Customer initiated Roll Over Instructions&lt;/li&gt; &lt;li&gt;Split Loan Instructions&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Close Loan Account Closure Request, if raised already.&lt;/li&gt; &lt;li&gt;Log an Online Alert Activity.&lt;/li&gt; &lt;/ol&gt; &lt;/p&gt;</p> <p>@param sessionId @param eventId @param accountId @param transactionDetailsDTO @param componentPreferenceDTO @param feeCollectionDTO @return @throws FatalException</p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method is responsible for deriving Periodic Fee Amount. @param applicationContext @param accountId @param eventDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method will fetch Pending periodic Fee for for account closure. @param applicationContext @param accountId @param eventDate @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method provides the service of EFS penalty charging. @param applicationContext @param accountId @param accountCurrencyCode @param penaltyAmount @param valueDate @param eventId
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	This method will generate a Payout Quote for an existing loan of the customer on the basis of loan Event Type. Quotes can be maintained for Partial Payoff(PPF)/Final Settlement(EFS)/Rate Change(Loan Event Type). A unique quote ID will be generated for each Quote generation. @param sessionContext @param loanEventSimulationDTO @return LoanSaveEarlyPayoutQuoteResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountClosure	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	<pre>This method is exposed to create pricing recommendations for a loan account. A pricing recommendation contains the applicable rates to the loan account for the specific periods based on the loan facts which are passed as inputs in loanAccountConfigurationDTO. The methods calls the {LoanPricingApplicationAdapter#createPricingRecommendation()} to generate a pricing response which contains the pricing ticket number. Pricing ticket number is reference to which all the fees and prices for a given account number are saved in pricing tables. </pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, String, arrangementNumber, 1..16,It indicates the loan arrangement number on which inquiry is to be done.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , This class holds the attributes pertaining to loan account configuration. @return {@link PricingResponse} @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	<pre>This method will be used by origination module to refresh the rates on offer Acceptance. It will update the Contract date with the offer acceptance date in loan account configuration ({@link com.ofss.fc.domain.loan.entity.account.LoanAccountConfiguration}). Create a new pricing Recommendation as on the contracted date which will return the pricing ticket no. pricing ticket no will be updated in loan account configuration ({@link com.ofss.fc.domain.loan.entity.account.LoanAccountConfiguration}). </pre> @param sessionContext @param arrangementNumber

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param loanAccountConfigurationDTO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	This method is used to create a pricing recommendation plan @param sessionContext @param arrangementNumber @param loanAccountConfigurationDTO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	This method is used to create a pricing recommendation plan @param sessionContext @param arrangementNumber @param loanAccountAmendConfigurationDTO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	This method is used to create a pricing recommendation plan @param sessionContext @param arrangementNumber @param loanAccountConfigurationDTO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendation	@param applicationContext @param arrangementNumber @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendationForCalculator	@param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @return @throws FatalException	This method is used to create a pricing recommendation plan @param sessionContext @param arrangementNumber @param loanAccountConfigurationDTO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	repriceIndexes	This method performs the repriceIndexes operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param repriceDate @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException	This method is used to repriceIndexes. @param applicationContext @param accountId @param repriceFormDTO @param rolloverDate @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	repriceIndexes	This method performs the repriceIndexes operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param repriceDate @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException	@param applicationContext @param arrangementNumber @param repriceFormDTO @param repriceDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForRecommendation		<p>This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. &lt;ul&gt; &lt;li&gt;If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.&lt;/li&gt; &lt;li&gt;fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.&lt;/li&gt; &lt;/ul&gt; @param applicationContext @param accountId @param effectiveDate @param loanInterestType @param repaymentStages @param offerId @param pricingTicketNumber @param productId @param fixedTermInMonths @param newApprovedAmount @param newLoanTerm @return SortedMap&lt;LoanInterestType, List&lt;NetRateDTO&gt;&gt;</p>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForRecommendation		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForRecommendation		<p>&lt;p&gt;This method is responsible for fetching Rates using Loan Account Configuration details. &lt;br&gt; It uses Loan Account Opening details from Loan Account Configuration details for preparing facts. &lt;/p&gt; @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendationForVariationRequest	@param applicationContext @param arrangementNumber @param variationFactDTO @return PricingResponse @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferingsForVariationRequest	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferingsForVariationRequest	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	<pre>This method is used to restructure the loan account based on the new schedule code &lt;br&gt; It has the following input parameters:- &lt;br&gt; 1)LoanOfferSwapDetailsDTO The following steps are carried out in this method:- &lt;br&gt; 1)The doRestructure() method of LoanRepaymentScheduleApplicationService is called with the schedule code, account number &amp; reschedule date i.e., process date. &lt;br&gt; </pre> @param loanOfferSwapDetailsDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferingsForVariationRequest	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	This method performs the operation of disbursing the parent Loan Account with the new sanctioned amount @param applicationContext @param parentAccountId @param totalPaidOffAmount @param processDate @param feeCollectionDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	saveAllOfferingsForVariationRequest	This method saves all the offerings of pricing module and creates a pricing plan in the pricing module. @param context @param accountId @param effectiveDate @throws FatalException	<pre>This method is used to restructure the loan account based on the new schedule code &lt;br> It has the following input parameters:- &lt;br> 1)LoanOfferSwapDetailsDTO The following steps are carried out in this method:- &lt;br> 1)The doRestructure() method of LoanRepaymentScheduleApplicationService is called with the schedule code, account number &amp; reschedule date i.e., process date. &lt;br> &lt;/pre> @param loanOfferSwapDetailsDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForVariationRequest	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param variationFactDTO @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param variationFactDTO @return LoanFactInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForVariationRequest	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param variationFactDTO @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param offerSwapDetailsDTO @return LoanFactInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForVariationRequest	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param variationFactDTO @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param loanConsolidationInputDTO @return LoanFactInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchFeeForPriceTicket	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it.   The arrangement number is a unique identifier used by modules to identify their respective entities.  For eg:CASA,TD,Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number.  This method will return all the fees for a given ticket number in the form of price feature DTOS {@link PriceFeatureDTO}   @param applicationContext @param priceTicketNumber @param currencyCodes @return an array of type {@link WrappedPriceFeatureDTO}	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it.   The arrangement number is a unique identifier used by modules to identify their respective entities.  For eg:CASA,TD,Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number.  This method will return all the fees for a given ticket number in the form of price feature DTOS {@link PriceFeatureDTO}   @param applicationContext @param priceTicketNumber @param currencyCodes @return an array of type {@link WrappedPriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an exception



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException if any of the arguments are invalid or if the { <a href="#">@link EventPriceService</a> } throws an exception	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRatesForDocumentGeneration	This method is responsible for fetching Rates required for Document Generation. @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return	<p>This method is responsible for providing the details required for Document Generation.   It uses Loan Schedule Configuration and Loan Account Configuration to derive the Rate details and Repayment details for every rate details. </p> @param sessionContext @param arrangementId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRatesForDocumentGeneration	This method is responsible for fetching Rates required for Document Generation. @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return	This method performs the operation of fetching the simulated Rates from Pricing . The rates returned from the Pricing module are simulated and not persisted internally by the Pricing module. <ul> <li>If the input repaymentStages is null, then the existing details for the stages are fetched via the LoanRepaymentScheduleRepository, else the supplied repaymentStages are used.</li> <li>fetchSimulatedRates is called through loanPricingApplicationAdapter and the return is NetRateDTO.</li> </ul> @param applicationContext @param loanAccountOpeningDTO @param effectiveDate @param loanInterestType @param repaymentStages @param fixedTermInMonths @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchSimulatedRatesForDocumentGeneration	This method is responsible for fetching Rates required for Document Generation. @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return	<p>This method is responsible for fetching Rates using Loan Account Configuration details.   It uses Loan Account Opening details from Loan Account Configuration details for preparing facts. </p> @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	<pre>This method is called from Bundle module to update the BundleBenefitApplicable for the account. @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	The method is responsible to inform bundle of any variations on the loan account attached to the bundle. Steps to be followed:<\n> 1 - Check if loan account passed has any bundle benefits.<\n> 2 - If yes, link bundle benefits to the arrangement/account post variation.<\n> 3 - Update pricing of the latest benefit status (All, Interest, Fee, None). @param applicationContext @param accountId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	This method notifies the linkage of an account to bundle. @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	This method is used to notify loans when there is a delinkage of an account from a bundle it was attached to. @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	This is the method to notify loans when the bundle attached to a loan account has expired. @param accountId @param bundleId @param bundleReferenceId @param isBundleBenefitApplicable @param bundleExpiryDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	This method is to notify loans when the loan account has been unsubscribed from the bundle it is attached to. @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	@param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param asOnDate @param fixedBenefitAppplicationStatus @param variableBenefitAppplicationStatus @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateBundleBenefitApplicationStatus	This method is responsible for notifying pricing about Bundle. @param applicationContext @param arrangementNumber @param benefitApplicabilityDTOArray @return	<p>This method is responsible for notifying BundleBenefitApplicableStatus.  This method is called whenever account is opened with offer bundle and also when bundle offer is delinked from account.</p> @param applicationContext @param loanPricingDetailsInputDTO @param loanAccountOpeningDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicFeeRefundDetailsWithPeriodicDates	This method fetches a List of {@link PeriodicFeeRefundFeatureDTO} from the pricing module indicating the current refund fee amount and the next computation date for obtaining refund. @param pricingEventContextDTO @param accountId @param refundDate @return List<PeriodicFeeRefundFeatureDTO>	This method credits the loan account with the fee refund amount (obtained from pricing module) and logs a fee  refund action for the next refund date (obtained pricing module)   It does the following,  <ol> <li>Call Pricing Service to fetch list of fees to be refunded for the current date (i.e refund fee amount and next refund date for each fee) via {@link LoanPricingApplicationAdapter.getPeriodicFeeRefundDetailsWithPeriodicDates}</li> <li>For each fee to be refunded</li> </ol> <blockquote> {  //Credit the loan account i.e RPA balance and also perform appropriation via {@link  LoanRefundFeesApplication#creditRpaAndAppropriate}  //log an action for next date, when fees are to be refunded via {@link LoanPricingFeeApplication#logLoanRefundFeesAction}  }</blockquote> @param applicationContext @param accountId @param actionDate @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicFeeRefundDetailsWithPeriodicDates	This method fetches a List of {@link PeriodicFeeRefundFeatureDTO} from the pricing module indicating the current refund fee amount and the next computation date for obtaining refund. @param pricingEventContextDTO @param accountId @param refundDate @return List<PeriodicFeeRefundFeatureDTO>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicFeeRefundDates	@param applicationContext @param accountId @return @throws FatalException	@param applicationContext @param accountId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicFeeRefundDates	@param applicationContext @param accountId @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	linkBundleBenefitsToArrangement	This method will stamp bundle variances to account. @param applicationContext @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate @throws FatalException	<pre>This method is called from Bundle module to update the BundleBenefitApplicable for the account. @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	linkBundleBenefitsToArrangement	This method will stamp bundle variances to account. @param applicationContext @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate @throws FatalException	The method is responsible to inform bundle of any variations on the loan account attached to the bundle. Steps to be followed:<\n> 1 - Check if loan account passed has any bundle benefits.<\n> 2 - If yes, link bundle benefits to the arrangement/account post variation.<\n> 3 - Update pricing of the latest benefit status (All, Interest, Fee, None). @param applicationContext @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	linkBundleBenefitsToArrangement	This method will stamp bundle variances to account. @param applicationContext @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate @throws FatalException	This method notifies the linkage of an account to bundle. @param accountId @param bundleId @param bundleReferenceId @param isNominatedForBundleFee @param isBundleBenefitApplicable @param bundleEffectiveEndDate @param asOnDate @param fixedBenefitApplicationStatus @param variableBenefitApplicationStatus @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	linkBundleBenefitsToArrangement	This method will stamp bundle variances to account. @param applicationContext @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricesSimulation	This method returns the simulated periodic fees @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricingSimulation	This method returns the simulated periodic fees @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricingSimulation	This method returns the simulated periodic fees @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricingSimulation	This method returns the simulated periodic fees @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricingSimulation	This method returns the simulated periodic fees @param applicationContext @param accountId @param eventCode @param evaluationDate @param startDate @param endDate @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getRepricingConfiguration	This method performs the getRepricingConfiguration operation @param accountId @param emptyForm @param repriceDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getRepricingConfiguration	This method performs the getRepricingConfiguration operation @param accountId @param emptyForm @param repriceDate @return @throws FatalException	This method performs the task of filling the repriceFormDTO so as to indicate to Pricing as to in what manner the account is to rePriced/refreshed. Depending on the return form dto, Pricing will rePrice/refresh the account accordingly. This method first checks whether the RateChange is eligible or not by calling the isRateChangeEligible() method. If it is eligible, it logs the corresponding RateChange action in an actions table, from which, the action would be picked up during the Batch for processing. If it is not eligible, then the RejectedEntry details are logged in a <RejectedEntry> table for internal audit purposes. Based on the eligibility of the Ratechange action & the rules from the <RepriceRulesTable>, the repriceFormDTO will be filled & returned. @param accountId @param repriceDate @param emptyForm @return RepriceFormDTO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getRepricingConfiguration	This method performs the getRepricingConfiguration operation @param accountId @param emptyForm @param repriceDate @return @throws FatalException	This method performs the task of filling the repriceFormDTO so as to indicate to Pricing as to in what manner the account is to rePriced/refreshed. Depending on the return form dto, Pricing will rePrice/refresh the account accordingly. This method first checks whether the RateChange is eligible or not by calling the isRateChangeEligible() method. If it is eligible, it logs the corresponding RateChange action in an actions table, from which, the action would be picked up during the Batch for processing. If it is not eligible, then the RejectedEntry details are logged in a LoanRateChangeRejectedAction Table table for internal audit purposes. Based on the eligibility of the RateChange action , the repriceFormDTO will be filled & returned. @param accountId @param repriceDate @param repriceFormDTO @return RepriceFormDTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForStageWithDateFilter	This method returns the net rate for account for the given stage and date. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @param dateToFilterOn @return	This method returns the net rate for account for the given stage and date. @param applicationContext @param arrangementNumber @param effectiveDate @param stage @param loanInterestType @param dateToFilterOn @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	computePeriodicFeeDates	This method will get call from Loans on disbursement date to get the periodic fee dates information if there is any periodic event associated with the pricing plan @param applicationContext @param accountId @param startDateForPeriodicCycle @return @throws FatalException	This method inserts new account schedule and blown up schedule. Also new schedule gets inserted into the schedule history tables with an uuid generated through FCRJUuidGenerator. Generated uuid will get stored in the LoanEvent which will be used for replay event in backdating process. @param applicationContext @param accountId @param onDate @param blownupSchedule @param scheduleIdForPeriodicFees TODO @param schedule
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	computePeriodicFeeDates	This method will get call from Loans on disbursement date to get the periodic fee dates information if there is any periodic event associated with the pricing plan @param applicationContext @param accountId @param startDateForPeriodicCycle @return @throws FatalException	This method will get call from Loans on disbursement date to get the periodic fee dates information if there is any periodic event associated with the pricing plan @param applicationContext @param accountId @param startDateForPeriodicCycle @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method will merge the rates during processing of rollover. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method will merge the rates during rollover simulation. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRates	This method performs the restructureRates operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRatesSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method will merge the rates during rollover simulation. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRatesSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRatesSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRatesSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureRatesSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendation	This method returns the simulated periodic fees when accountId is not available during the recommendation mode. @param applicationContext @param evaluationDate @param startDate @param endDate @param accountCurrencyCode @param pricingTicketNumber @param sanctionedAmount @return @throws FatalException	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForAccountClosure	This method derives all facts for a given loan account in case of Loan Account Closure using "Early & Final Settlement" @param sessionContext @param accountId @param processDate @param transactionAmountForfacts @return PricingEventContextDTO @throws FatalException	<p>This method is responsible for deriving all facts for a Loan Account provided in case of Loan Account Closure via <b>"Early & Final Settlement"</b> exclusively.   It derives the Total Due Amount, which is being considered as BASE Amount for deriving Fees.   As a part of inquiry, Fee amounts gets calculated and added to Total Due Amount.  Hence it is necessary to deduct the fee amounts from the Total Due amount, so that Pure Due amount (without Fees) can be used for fetching Fees on User Interface. </p> @param sessionContext @param accountId @return LoanFactInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForAccountClosure	This method derives all facts for a given loan account in case of Loan Account Closure using "Early & Final Settlement" @param sessionContext @param accountId @param processDate @param transactionAmountForfacts @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of Loan Account Closure using "Early & Final Settlement" @param sessionContext @param accountId @param processDate @param transactionAmountForfacts @return PricingEventContextDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	closePricingPlan	@param arrangementNumber @throws FatalException	<p>This method will close the loan account on the inputed date.  Following actions are being performed by this method:-  <ol> <li>Update Loan Account status as "Closed"</li> <li>Update Loan Account Closure Date</li> <li>Update reason for Loan Account Closure</li> <li>Update Central Bank Report details</li> <li>Update Bundle Subscription of Loan Account</li> <li>Update Asset Classification</li> <li>Delink Loan Account from Facility</li> <li>Remove Offset Linkages</li> </ol></p> @param applicationContext @param account @param processDate @param eventId TODO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	closePricingPlan	@param arrangementNumber @throws FatalException	This method closes the loan account. @param applicationContext @param accountId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	closePricingPlan	@param arrangementNumber @throws FatalException	This method will be used to unregister or close a pricing plan. If the account is being closed, the pricing plan linked to the account needs to be phased out. The method will in turn call the service {link ContractApplicationService#closePricingPlan} to close the pricing plan. @param arrangementNumber @param pricingContractContextDTO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForOnlineRequest	This method derives all facts for a given loan account in case of online request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of online events like Loan Disbursement and details set in loanAccountOnlineEventsDTO. @param sessionContext @return LoanFactInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForOnlineRequest	This method derives all facts for a given loan account in case of online request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForOnlineRequest	This method derives all facts for a given loan account in case of online request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getLoanFactsDTOForOnlineRequest	This method derives all facts for a given loan account in case of online request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when an variation request is raised for account through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param applicationContext @param loanAccountVariationDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getFeesForTicketNumber	This method derives all facts for a given loan account in case of online events like Loan Disbursement id and details set in loanAccountOnlineEventsDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method returns the fee applicable for a price ticket number @param sessionContext @param accountId @param pricingTicketNumber @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getFeesForTicketNumber	This method derives all facts for a given loan account in case of online events like Loan Disbursement id and details set in loanAccountOnlineEventsDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices, evaluates them to obtain objects of the type { <a href="#">ProcessedBasePriceDTO</a> }, encapsulates them into a { <a href="#">PriceFeatureDTO</a> } and returns it. The arrangement number is a unique identifier used by modules to identify their respective entities. For eg:CASA,TD,Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number. This method will return all the fees for a given ticket number in the form of price feature DTOS { <a href="#">PriceFeatureDTO</a> } @param applicationContext @param priceTicketNumber @param currencyCodes @return an array of type { <a href="#">WrappedPriceFeatureDTO</a> } @throws FatalException if any of the arguments are invalid or if the { <a href="#">EventPriceService</a> } throws an exception
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getFeesForTicketNumber	This method derives all facts for a given loan account in case of online events like Loan Disbursement id and details set in loanAccountOnlineEventsDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException	This method derives all facts for a given loan account in case of online events like Loan Disbursement id and details set in loanAccountOnlineEventsDTO. @param sessionContext @return PricingEventContextDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesSimulationAtRecommendationForBundle		This method returns the simulated Periodic Prices during recommendation, when accountId is not available.And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricesSimulationForBundle		This method generates the feeSchedule simulation for the input repaymentStages @param applicationContext @param scheduleId @param onDate @param repaymentStages @param isLoanInRecommendation TODO @param pricingTicketNumber TODO @param bundleId TODO @param bundleReferenceId TODO @param isAccountNominatedForBundlePredoicFee TODO @param accountScheduleAttributesDTO TODO @param temp TODO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricesSimulationForBundle		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getDummyPeriodicPricesSimulationForBundle		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalToPricing		This method is used for fee reversal processing. @param applicationContext @param accountId @param transactionDetailsDTO @param feeCollectionDTO
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	intimateReversalToPricing		This method is used for fee reversal processing. @param applicationContext @param accountId @param transactionDetailsDTO @param feeCollectionDTO @param eventId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updatePricingRecommendation	This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updatePricingRecommendation	This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when an variation request is raised for account through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param applicationContext @param loanAccountVariationDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchFeesOnEventForRecommendation		The method is responsible to fetch fees for an event in Reco mode. @param applicationContext @param accountConfigurationDTO @param loanEventCodes @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchFeesOnEventForRecommendation		The method is responsible to fetch fees for an event in Reco mode. @param applicationContext @param accountConfigurationDTO @param loanEventCodes @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchFeesOnEventForRecommendation		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		This method is used to charge and capitalize the total fees accrued in the moratorium period on moratorium stage end date. @param accountId @param currencyCode @param compoundDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		<pre> This method is responsible for management of interactions with pricing module for prices and levying the fees on the account Fees are defined in pricing module for different period Typical interaction with pricing module will involve a)Fetching of Fees for a period b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete The inputs to the method are :- a)ApplicationContext b)accountId c)eventId d)List&lt;PriceFeatureDTO&gt; e)DeductionModeType (Bill , Deduct , Capitalize) @param applicationContext @param accountId @param deductionModeType @param wrappedPriceFeatureDTO @param deductionModeType </pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		<pre> This method is used for processing the online Fees: Fees are defined in pricing module for list of module events Typical interaction with pricing module will involve a)Fetching of Fees for an event b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete @param applicationContext @param accountId @param eventId @param wrappedPriceFeatureDTO @param deductionModeType @param transactionAmount @param pricingTicketNumber @param priceFeatureDTO @param accountCurrency @throws FatalException </pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		<pre> This method will convert from priceFeatureDTO array to WrappedPriceFeatureDTO array. @param priceFeatureDTO @return </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		The method will levy pending facility fees on account closure and also inform LCM on account closure @param applicationContext @param accountId @param valueDate @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		The method will return pending facility fees as on account closure date @param accountId @param valueDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		This method is used to get the details of the deferred fees @throws FatalException @ accountId
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		Given an event and an account/arrangement number, this method retrieves the codes of the applicable prices,   evaluates them to obtain objects of the type {@link ProcessedBasePriceDTO}, encapsulates them into a {@link PriceFeatureDTO} and returns it.   The arrangement number is a unique identifier used by modules to identify their respective entities.  For eg:CASA,TD,Lending would use account number as the arrangement number and Origination would probably use application id as the arrangement number.  This method will return all the fees for a given ticket number in the form of price feature DTOS {@link PriceFeatureDTO}   @param applicationContext @param priceTicketNumber @param currencyCodes @return an array of type {@link WrappedPriceFeatureDTO} @throws FatalException if any of the arguments are invalid or if the {@link EventPriceService} throws an

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				exception
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fromPriceFeatureDTOToWrappedPriceFeatureDTO		This method would fetch the fees capitalized by origination. @param arrangementNumber @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchCapitalizedFees	This method fetches capitalized fees. @param accountId @return @throws FatalException	<pre> This method is responsible for management of interactions with pricing module for prices and levying the fees on the account Fees are defined in pricing module for list of module events Typical interaction with pricing module will involve a)Fetching of Fees for an event b)Levying of fees on the account c)Intimating pricing with information of fee levied on the account d)Notifying pricing that a fee transaction is complete The inputs to the method are :- a)ApplicationContext b)accountId c)eventId d)List&lt;PriceFeatureDTO&gt; e)DeductionModeType (Bill , Deduct , Capitalize) It is expected that fee taskflow embeded in screens will have set the values correctly in PriceFeatureDTO in relation to the negotiated amount and waived amount If(eventId == LOAN_DISBUSREMENT) then invoke manageDisbusrementFees() else Iterate through the PriceFeatureDTO do the following :- 1)Get the fee amount to be leveied , fee amount to waived in a local variable. 2)If feeAmountToBeLevied &gt; ZERO &lt;br&gt; If DeductionModeType == BILL_FEES then invoke arrear.raise and pass the required details invoke intimatePricing and pass the required details If DeductionModeType == DEDUCT_FEES then &lt;br&gt; invoke arrear.raiseAnPay and pass the required details invoke intimatePricing and pass the required details If DeductionModeType == </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				CAPITALIZE_FEES then invoke arrear.capitalize and pass the required details invoke intimatePricing and pass the required details 3)If feeAmountToBeWaived > ZERO   invoke Arrear.record and pass the required details &#064;param applicationContext &#064;param accountId &#064;param wrappedPriceFeatureDTO @param deductionModeType
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchCapitalizedFees	This method fetches capitalized fees. @param accountId @return @throws FatalException	<p>The method is responsible to levy deferred fees during hardship relief period. This method will levy following deffered fees <ul> <li>Loan Periodic fees</li> <li>Penalty Fees</li> <li>Facility Fees</li> <li>Bundle maintenance fees</li> </ul>  @param applicationContext @param accountId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchCapitalizedFees	This method fetches capitalized fees. @param accountId @return @throws FatalException	@param applicationContext @param eventId @param accountId @param disbursementAmount @param loanAccount @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchCapitalizedFees	This method fetches capitalized fees. @param accountId @return @throws FatalException	This method would fetch the fees capitalized by origination. @param arrangementNumber @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateArrangementIdToAccountIdForFees	Update arrangementId to accountId for fees. @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException	This method would mark fees as capitalized in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateArrangementIdToAccountIdForFees	Update arrangementId to accountId for fees. @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException	This method would mark fees as deferred in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPeriodicPricesWithCapitalizedPrices	Fetches fees for periodic event like moratorium, hardship etc. @param sessionContext @param factDTO @param arrangementNumber @param evaluationDate @param currencyCodes @param computationDate @return @throws FatalException	Fetch Periodic fees for MOR, Hardship stage etc. @param sessionContext @param context @param accountId @param evaluationDate @param currencyCodes @param computationDate @param amount @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	markFeeTransactionAsComplete	Given an ticket Number this method will load all the Fees for that ticketNumber {@link FlattenedFeeTransactionDetails} and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException	This method would mark fees as capitalized in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	markFeeTransactionAsComplete	<p>Given an ticket Number this method will load all the Fees for that ticketNumber {@link FlattenedFeeTransactionDetails} and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException</p>	<p>This method would mark fees as deffered in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	markFeeTransactionAsComplete	<p>Given an ticket Number this method will load all the Fees for that ticketNumber (<a href="#">@link FlattenedFeeTransactionDetails</a>) and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException</p>	<p>This method is responsible for update pricing that a transaction pertaining to fees is completed. &lt;br&gt; This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction &lt;br&gt; @param pricingTicketNumber @param eventCodes</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	markFeeTransactionAsComplete	Given an ticket Number this method will load all the Fees for that ticketNumber {@link FlattenedFeeTransactionDetails} and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquireDeferredRepriceActions	This method will fetch all deferred actions for a given account and priceType. @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException	This method will fetch all deferred actions for a given account and priceType. @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePendingDeferredRepriceActions	This method will fetch all pending deferred actions for a given account and priceType (returns actions with status NOT_STARTED). @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException	This method will fetch all pending deferred actions for a given account and priceType (returns actions with status NOT_STARTED). @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePendingDeferredRepriceActions	This method will fetch all pending deferred actions for a given account and priceType (returns actions with status NOT_STARTED). @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException	The method will be used log a Rate change action for variable period start date and will be marked as Deferred till variable period start date. @param accountId @param repriceFormDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	inquirePendingDeferredRepriceActions	This method will fetch all pending deferred actions for a given account and priceType (returns actions with status NOT_STARTED). @param sessionContext @param accountId @param priceType @param triggeringFactCode @return @throws FatalException	The method will be used log a Rate change action for variable period start date and will be marked as Deferred till variable period start date. @param accountId @param repriceFormDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	removeDeferredRepriceActions	The method will remove a deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	The method will remove a deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	removeDeferredRepriceActions	The method will remove a deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	The method will be used log a Rate change action for variable period start date and will be marked as Deferred till variable period start date. @param accountId @param repriceFormDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	removeDeferredRepriceActions	The method will remove a deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	The method will be used log a Rate change action for variable period start date and will be marked as Deferred till variable period start date. @param accountId @param repriceFormDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	processDeferredRepriceActions	The method will process the deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	The method will process the deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	processDeferredRepriceActions	The method will process the deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	This method processes the Rate change which has been initiated by the Pricing module. @param applicationContext @param loanRateChangeAction @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	processDeferredRepriceActions	The method will process the deferred reprice action. @param sessionContext @param accountId @param priceType @param triggeringFactCode @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesAsOnDate	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return	This method will merge the rates during processing of rollover. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesAsOnDate	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return	This method performs the operation of fetching the existing rates from the pricing module as on Back value date, from the rates history.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository as on back value date from history otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param accountId @param valueDate @param loanInterestType @param stageNames @return SortedMap<LoanInterestType, List<NetRateDTO>>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesAsOnDate	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesAsOnDate	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return	This method will fetch rates from History which are already persisted for given back value date. @param applicationContext @param accountId @param asOnDate @param stage @param loanInterestType @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	<p>This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> <pre>@param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]</pre></p>	<pre>&lt;pre&gt; This operation is responsible for opening a Loan Account as well configuration of Preferences and facilities associated with a Loan Account. They are :- (1)Account Preferences (2)Accelerated Payment Instruction (3)Offset Account (4)Account Payment Instruction (5)Repayment Schedule Configuration (6)Rate Lock (7)Loan Account Details These details are captured by the Origination system and the details are passed to Loans module as part of the "Account Creation" process.The method works as follows :- (1)Invoke openLoanAccount method which will create a Loan Account (2)Fetch the AccountPreference details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke LoanAccountPreferenceApplicationService to save the Account Preference (3)Fetch the Accelerated Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAcceleratedPaymentInstructionDTO. Invoke LoanAcceleratedPaymentInstructionApplicationService to save the LoanAcceleratedPaymentInstruction (4)Fetch the OffsetAccount details from LoanAccountConfigurationDTO Set the accountId in the OffsetAccountDTO. Invoke OffsetAccountApplicationService to save the OffsetAccount (5)Fetch the Account Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke Account Payment InstructionApplicationService to save the Account Payment Instruction (6)Fetch the Repayment Schedule Configuration details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke</pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanRepaymentScheduleApplicationService to save the Repayment Schedule Configuration <code>&lt;/pre&gt;</code>  <code>@fcb.param MI,SessionContext,sessionContext, ,</code> This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. <code>@fcb.param MI,LoanAccountConfigurationDTO,loanAccountConfigurationDTO.</code> This holds the details required when opening and configuring a loans account. The transaction status will indicate whether the transaction went through successfully and causes the error if unsuccessful. <code>&lt;b&gt;Success</code> <code>&lt;li&gt;</code> <code>{@link com.ofss.fc.common.FAPIErrorConstants#MID_SUCCESS}&lt;/li&gt;</code> <code>&lt;b&gt;Transaction failure</code> Error Reason may be one of the following <code>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EBA_PROD_NOTMAPPED}</code> <code>&lt;/li&gt;</code> <code>&lt;li&gt;</code> <code>{@link com.ofss.fc.common.FAPIErrorConstants#MID_EGL_TXN_NO_INV}&lt;/li&gt;</code> <code>&lt;/ul&gt;</code> <code>@return</code> <code>{@link com.ofss.fc.app.loan.dto.account.LoanAccountOpeningResponse}</code> This response object holds information of the newly generated account and the status of transaction. <code>@throws FatalException</code> in case of error.Please refer to the documentation above for <code>&lt;i&gt;Transaction failure&lt;/i&gt;</code> section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]	<pre>&lt;pre&gt;This method is responsible for saving the account opening configuration done in Origination Systems. This method saves the following configuration in Loans: (1) Account Preference (2) Offset Account configuration (3) Account Payment Instruction (4) Repayment Schedule Configuration (5) Accelerated Payment Instruction (6) Rate Lock configuration Before saving the details, the following steps are first executed: (1) Validate account configuration w.r.t Product setup (2) Validate account configuration w.r.t Offer level parameters (3) Validate account configuration w.r.t Facility level setup On successful completion of the transaction, account configuration will be saved along with pricing recommendation. &lt;/pre&gt;</pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , Session Context which must contain the session ticket and user id.  @return {@link SaveLoanAccountConfigurationResponse}. @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param asOnDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			effectiveDate @return ContractedNetRateDTO[]	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	This method this will persist the contracted rates as of given date. And to be called during : <pre><li> Account Configuration, after creating recommendation</li></pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]	This method will persist Contracted rate for Rate lock in recommendation Mode. @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	This method this will persist the contracted rates as of given date. And to be called during : <pre><li> Account Configuration, after creating recommendation</li></pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]	This method is used to persist contracted net rates for recommendation. @param sessionContext @param pricingContextDTO @param stageDetails @param rateContractType @param asOnDate @return NetRateInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRatesForRecommendation	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt;</pre> @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return	<pre>&lt;pre&gt;</pre> This operation is responsible for opening a Loan Account as well configuration of Preferences and facilities associated with a Loan Account. They are :- (1)Account Preferences (2)Accelerated Payment Instruction (3)Offset Account (4)Account Payment Instruction (5)Repayment Schedule Configuration (6)Rate Lock (7)Loan Account Details These details are captured by the Origination system and the details are passed to Loans module as part of the "Account Creation" process.The method works as follows :- (1)Invoke openLoanAccount method which will create a Loan Account (2)Fetch the AccountPreference details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke LoanAccountPreferenceApplicationService to save the Account Preference (3)Fetch the Accelerated Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAcceleratedPaymentInstructionDTO. Invoke LoanAcceleratedPaymentInstructionApplicationService to save the LoanAcceleratedPaymentInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>(4)Fetch the OffsetAccount details from LoanAccountConfigurationDTO Set the accountId in the OffsetAccountDTO. Invoke OffsetAccountApplicationService to save the OffsetAccount (5)Fetch the Account Payment Instruction details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke Account Payment InstructionApplicationService to save the Account Payment Instruction (6)Fetch the Repayment Schedule Configuration details from LoanAccountConfigurationDTO Set the accountId in the LoanAccountPreferenceDTO. Invoke LoanRepaymentScheduleApplicationService to save the Repayment Schedule Configuration</p> <pre>&lt;/pre&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,LoanAccountConfigurationDTO,loanAccountConfigurationDTO. This holds the details required when opening and configuring a loans account. The transaction status will indicate whether the transaction went through successfully and causes the error if unsuccessful. &lt;b&gt;Success &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_SUCCESS} &lt;/li&gt; &lt;b&gt;Transaction failure Error Reason may be one of the following {@link com.ofss.fc.common.FAPIErrorConstants#MID_EBA_PROD_NOTMAPPED} &lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.common.FAPIErrorConstants#MID_EGL_TXN_NO_INV} &lt;/li&gt; &lt;/ul&gt; @return {@link com.ofss.fc.app.loan.dto.account.LoanAccountOpeningResponse} This response object holds information of the newly generated account and the status of</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				transaction. @throws FatalException in case of error.Please refer to the documentation above for <i>Transaction failure</i> section
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre><li> Rate lock, after adding new rate lock quote.</li> </pre> @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return	<pre>This method is responsible for saving the account opening configuration done in Origination Systems. This method saves the following configuration in Loans: (1) Account Preference (2) Offset Account configuration (3) Account Payment Instruction (4) Repayment Schedule Configuration (5) Accelerated Payment Instruction (6) Rate Lock configuration Before saving the details, the following steps are first executed: (1) Validate account configuration w.r.t Product setup (2) Validate account configuration w.r.t Offer level parameters (3) Validate account configuration w.r.t Facility level setup On successful completion of the transaction, account configuration will be saved along with pricing recommendation. </pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , Session Context which must contain the session ticket and user id.  @return {@link SaveLoanAccountConfigurationResponse}. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param asOnDate @return @throws FatalException</pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method will persist Contracted rate for Rate lock in recommendation Mode. @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method this will persist the contracted rates as of given date. This can be used in both current dated and back dated. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method is used to persist contracted net rates for recommendation. @param sessionContext @param pricingContextDTO @param stageDetails @param rateContractType @param asOnDate @return NetRateInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Account Configuration, after creating recommendation&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param inputStageDetailsDTOs @param rateContractType @param effectiveDate @return ContractedNetRateDTO[]</pre>
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistContractedRates	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param rateContractType @param asOnDate @return</pre>	This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt; Rate lock, after adding new rate lock quote.&lt;/li&gt; &lt;/pre&gt; @param applicationContext @param loanPricingDetailsInputDTO @param asOnDate @return</pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param repaymentStages @param loanInterestType @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return SortedMap<Date, Money> feeSchedule
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod	This method simulates and returns back-dated rates. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return NetRateDTO[]

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method simulates and returns back-dated periodic Fees and rate. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return RateAndFeesInquiryResponse
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedFees	This method simulates and returns back-dated periodic Fees. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedRates	This method simulates and returns back-dated rates. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method simulates and returns back-dated rates. @param applicationContext @param accountId @param fromDate @param toDate @param repaymentStages @param loanInterestType @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchBackdatedSimulatedRates	This method simulates and returns back-dated rates. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedRatesAndFees	This method persists and returns back-dated periodic rate and Fees. @param applicationContext @param accountId @param fromDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedRatesAndFees	This method persists and returns back-dated periodic rate and Fees. @param applicationContext @param accountId @param fromDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method persists and returns back-dated periodic rate. @param applicationContext @param accountId @param fromDate @param repaymentStages @param loanInterestType @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getReversalFeeDetails	This method fetches reversal fee details for particular event based fee between dates. @param accountId @param backValueDate @param processDate @param string @return @throws FatalException	@see com.ofss.fc.app.adapter.loan.ILoanPricingApplicationAdapter#getReversalFeeDetails(java.lang.String, com.ofss.fc.datatype.Date, com.ofss.fc.datatype.Date, java.lang.String)
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getReversalFeeDetails	This method fetches reversal fee details for particular event based fee between dates. @param accountId @param backValueDate @param processDate @param string @return @throws FatalException	@param accountId @param backValueDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updatePricingRecommendationForVariation	This method is called to update RBM value(Rebate Benefit Margin) when an variation request is raised for account through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param applicationContext @param loanAccountVariationDTO @throws FatalException	This method is called to update RBM value(Rebate Benefit Margin) when an variation request is raised for account through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param applicationContext @param loanAccountVariationDTO @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method will merge the rates during processing of rollover. @param applicationContext @param accountId @param rolloverDate @param splitDate @param repriceFormDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEvents	@param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedSimulatedFeesAndRates		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedSimulatedFeesAndRates		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedSimulatedFeesAndRates		This method persists and returns back-dated periodic rate and Fees. @param applicationContext @param accountId @param fromDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedSimulatedFeesAndRates		This method persists and returns back-dated periodic rate and Fees. @param applicationContext @param accountId @param fromDate @param toDate @param considerVarianceChanges @param computeFeesForLapsedPeriod @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	persistBackdatedSimulatedFeesAndRates		This method persists and returns back-dated periodic Fees and rate. @param applicationContext @param accountId @param fromDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	updateLoanAmendPricingRecommendation		This method is called to update RBM value(Rebate Benefit Margin) when account is configured through broker or introducer. This method will update the RBM value in the pricing plan created during recommendation @param sessionContext @param arrangementNumber @param agentArrangementLinkageDTO @param pricingTicketNumber @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchFeesOnEventForRecommendationOnAmend	The method is responsible to fees for a given account recommendation and a given event id. This would be useful in case of new account configuration or some variation events wherein the account configuration for event fees are to be fetched do not yet have a pricing plan.	The method is responsible to fetch fees for an event in Reco mode. @param applicationContext @param accountConfigurationDTO @param loanEventCodes @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	createPricingRecommendationForAmend	@param applicationContext @param arrangementNumber @return @throws FatalException	This method is used to create a pricing recommendation plan @param sessionContext @param arrangementNumber @param loanAccountAmendConfigurationDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method derives all facts for a given loan account in case of lending variation request, depending on event id and details set in LoanAccountVariationDTO. @param sessionContext @param variationFactDTO @return LoanFactInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method derives all facts for a given loan account in case of online events like Loan Disbursement and details set in loanAccountOnlineEventsDTO. @param sessionContext @return LoanFactInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method returns the simulated Periodic Prices during recommendation, when accountId is not available. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method returns the simulated Periodic Prices during recommendation, when accountId is not available.And account is opened with Bundle Benefits. @param applicationContext @param evaluationDate @param startDate @param endDate @return @throws FatalException @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method fetches the periodic prices with capitalized prices.
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method simulates and returns back-dated periodic Fees and rate. @param applicationContext @param accountId @param fromDate @param toDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return RateAndFeesInquiryResponse
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchPeriodFactDTO	This method will fetch rate type configuration for an account and generate Pricing Fact information for the same. @param accountId @return SortedSet<PeriodFactInformation>	This method persists and returns back-dated periodic Fees and rate. @param applicationContext @param accountId @param fromDate @param loanPricingDetailsInputDTO @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountVariation		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountVariation		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountVariation		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getPriceForAccountVariation		@param applicationContext @param accountId @param eventId @param eventDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEventsSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException	This method performs the operation of fetching the existing rates from the pricing module.  <ul> <li>If the input stageNames is null then the existing stageNames are retrieved via LoanRepaymentScheduleRepository otherwise the supplied stageNames are used. </li> </ul> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return SortedMap<LoanInterestType, List<NetRateDTO>> @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEventsSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	restructureFeesAndRatesConsideringRepriceEventsSimulation	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException	This method performs the restructureRatesSimulation operation @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is used to fetch rate for a given stage and date combination. @param applicationContext @param arrangementNumber @param asOnDate @param installmentCalculationMethod @param pricingElementType @param dateToFilterOn @return NetRateDTO[]	This method is used to get reprice frequency from pricing module. @param applicationContext @param arrangementNumber @param installmentCalculationMethod @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForStageWithDateAndAmountFilter	This method is used to fetch rate for a given stage and date combination. @param applicationContext @param arrangementNumber @param asOnDate @param installmentCalculationMethod @param pricingElementType @param dateToFilterOn @return NetRateDTO[]	This method is used to fetch rate for a given stage and date combination. @param applicationContext @param arrangementNumber @param asOnDate @param installmentCalculationMethod @param pricingElementType @param dateToFilterOn @return NetRateDTO[]
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	deleteContractedRates		This method this will persist the contracted rates as of given date. And to be called during : <pre>&lt;li&gt;Account Configuration, after creating recommendation&lt;/li&gt;</pre> @param applicationContext @param arrangementNumber @param loanAccountConfigurationDTO @param asOnDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	deleteContractedRates		This method will delete the Contracted Net Rates and Net Rate Plan for given ticket number. @param sessionContext @param ticketNumber @throws FatalException @author ManishDix @since 2.2
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	deleteContractedRates		
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getNegotiatedOriginationFees	This method fetches the waived origination fees during EFS which needs to be charged or waived off. @param applicationContext @param accountNumber @return @throws FatalException	@param applicationContext @param accountId @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getNegotiatedOriginationFees	This method fetches the waived origination fees during EFS which needs to be charged or waived off. @param applicationContext @param accountNumber @return @throws FatalException	This method will doAccountCloser @param applicationContext @param account @param eventId @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForArrangementOnGivenAndPreviousDate	This method is used to fetch the rates for the input date and the input date - 1. @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException	This method is used to log the non-financial statement for net rate change. @param sessionContext @param accountId @param valueDate @param eventId @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	fetchRatesForArrangementOnGivenAndPreviousDate	This method is used to fetch the rates for the input date and the input date - 1. @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException	This method is used to fetch the rates for the input date and the day before. @param sessionContext @param arrangementNumber @param asOnDate @return NetRateStatementInquiryResponse @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	rearrangeDatesForRates	This method is used to fetch rates from pricing when maturity date is modified(e.g repayment day change). @param sessionContext @param accountId @param offerId @param productGroupId @param loanPricingDetailsInputDTO @param asOnDate @param isSimulation @return	This method is used to fetch rates from pricing when maturity date is modified(e.g repayment day change). @param sessionContext @param accountId @param asOnDate @param isPostMaturity @param isSimulation @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	rearrangeDatesForRates	This method is used to fetch rates from pricing when maturity date is modified(e.g repayment day change). @param sessionContext @param accountId @param offerId @param productGroupId @param loanPricingDetailsInputDTO @param asOnDate @param isSimulation @return	This method is used to fetch rates from pricing when maturity date is modified(e.g repayment day change). @param sessionContext @param accountId @param offerId @param productGroupId @param loanPricingDetailsInputDTO @param asOnDate @param isSimulation @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	adjustUDMEntriesForVariableStage	Refer Document - R2.3_LENDNG_Carry forward UDM - Backdated Offer Swap & Rollover_FS.docx This method is to send an adjustment entry for UDM if fixed period ends before old fixed end date due to rollover i.e. if user has done move to fixed for lesser period or move to variable with carry forward UDM flag true then old UDMs which are starting from old fixed end date should also be applied for newly added variable portion. @param applicationContext @param arrangementNumber @param newFixedEndDate @param oldFixedEndDate @param loanCalculationMethodtype @return @throws FatalException	This method performs the operation of processing the loan rollover instruction. <li>The stage as on the rollover date is obtained.</li> <li>Stage would be split on either repayment review Date or Last Charging Date.</li> <li> nextChargingDate is obtained which is after the given start date after which the given type of schedule event will occur and then Rollover Business Policy is applied.</li> <li>There are cases on which the loanStage are either elapsed or not.</li> <li>If RolloverType equals MOVE_TO_FIXED then four Cases are handled.</li>   Case 1: Stage s1 is partially in newFixedTerm i.e. s1 starts before newFixedTerm & ends in newFixedTerm  case:2 Stage s1 is partially in newFixedTerm i.e. s1 starts in newFixedTerm & ends after newFixedTerm  case3: Stage s1 is partially in newFixedTerm i.e. s1 starts before newFixedTerm & ends after newFixedTerm  case:4 Stage s1 is fully in newFixedTerm i.e. s1 starts in newFixedTerm & ends in newFixedTerm  <li>If fixedTermRange contains stageStartDate or fixedTermRange contains stageEndDate or fixedTermStartDate is after Or Equal to stageStartDate and fixedTermEndDate is Before Or Equal to stageEndDate</li> <li> AllowInterestRate is set to true.</li> <li>AllowMarginRate is set to true.RetainUDM is set to false. stageStartDate is in newFixedTermRange and processing is done.</li> <li>If RolloverType equals MOVE_TO_VARIABLE then four cases are handled.</li>   Case1: Stage s1 is partially in newVariableTerm i.e. s1 starts before newVariableTerm & ends in newVariableTerm  case2 : Stage s1 is partially in newVariableTerm i.e. s1 starts in newVariableTerm & ends after newVariableTerm  case 3: Stage s1 is partially in newVariableTerm i.e. s1 starts before

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>newVariableTerm &amp; ends after newVariableTerm&lt;br&gt;  case 4: Stage s1 is fully in newVariableTerm i.e. s1 starts in newVariableTerm &amp; ends in newVariableTerm&lt;br&gt; &lt;li&gt;If variableTermRange contains stageStartDate or variableTermRange contains stageEndDate or variableTermStartDate is after Or Equal stageStartDate and variableTermEndDate is before or equal stageEndDate AllowInterestRate is set to true AllowMarginRate is set to false RetainUDM is set true stageStartDate is checked whether it is in the newvariableTermRange Schedule fees is called and the service is called.&lt;/li&gt; &lt;li&gt; LoanInterestComputationService.logComputationActionOnLoanReschedule is called to recompute the interest for the schedule.&lt;/li&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	adjustUDMEntriesForVariableStage	Refer Document - R2.3_LENDNG_Carry forward UDM - Backdated Offer Swap & Rollover_FS.docx This method is to send an adjustment entry for UDM if fixed period ends before old fixed end date due to rollover i.e. if user has done move to fixed for lesser period or move to variable with carry forward UDM flag true then old UDMs which are starting from old fixed end date should also be applied for newly added variable portion. @param applicationContext @param arrangementNumber @param newFixedEndDate @param oldFixedEndDate @param loanCalculationMethodtype @return @throws FatalException	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getApplicableRateOnDate	This method is used to get the applicable rates on the current date for an array of net rate dtos. @param interestRates @param onDate @return	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> <li>FacilityId</li> <li>OutstandingAmount</li> <li>CurrentFixedRate</li> <li>QuoteExpiryDate</li> </pre> @param context @param accountId @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getApplicableRateOnDate	This method is used to get the applicable rates on the current date for an array of net rate dtos. @param interestRates @param onDate @return	This method returns the interest rates applicable on the current process date in form of {@link NetRateDTO}. @param applicationContext @param accountId @return netRateDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getApplicableRateOnDate	This method is used to get the applicable rates on the current date for an array of net rate dtos. @param interestRates @param onDate @return	This method is used to get the margin for logging it in the statement. @param applicationContext @param accountId @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getApplicableRateOnDate	This method is used to get the applicable rates on the current date for an array of net rate dtos. @param interestRates @param onDate @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getApplicableRateOnDate	This method is used to get the applicable rates on the current date for an array of net rate dtos. @param interestRates @param onDate @return	
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getMaxMargin		@param sessionContext @param serviceId @param requestDTO @param responseDTO @return @throws FatalException
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getMaxMargin		@param pricingTicketNumber @return
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	getMaxMargin		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanPricingApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanPricingApplicationAdapter	validatePriceChange	@param applicationContext @param priceChangeDate @param accountId @throws FatalException	Validation for Loan accounts @param eventPriceVarianceDTOArray @param arrangementNumber @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	Validate Add on card holder. This validation method does the below validations,  <ul> <li>Age of add on card holder not less than the minimum age maintained in the bank's policy parameter</li> <li>Add on card holder is allowed for this offer check</li> <li>Validate Maximum add-on cards for this offer</li> </ul> @param sessionContext @param addOnCardHolderDTOs @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	The method fetchBillCode is used to fetch the bill code based on the derivation fact provided.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BillCodeDerivationRequestDTO,billCodeDerivationRequestDTO, ,This attribute contains account id, credit card account configuration dto and bundle id. @return {@link CreditCardBillCodeResponse} This class contains the information related to credit card account details. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	The method inquireCardAccountDetails is used to fetch credit card account details for the given card account number.   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, This attribute denotes the card account number. @return {@link CreditCardAccountDetailsResponse} This class contains the information related to credit card account details. @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	Populate Credit Card Account DTO based on the provided input @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	Populate Credit Card Account DTO based on the provided input @param sessionContext @param creditCardAccountInputDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	Fetch offer name from PMU @param sessionContext @param creditCardAccountDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	This method is used to create credit facility using the external credit card details @param sessionContext @param creditCardDetailsDTO @param borrowingEntityId @return CreditFacilityDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionId @param offerId @return OfferInquiryServiceResponse @throws FatalException	Populate Credit Card Details Post Inquiry @param sessionId @param creditCardDetailsDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionId @param offerId @return OfferInquiryServiceResponse @throws FatalException	Populate Credit Card Account DTO from credit card details DTO @param sessionId @param creditCardDetailsDTO @return CreditCardAccountDTO @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionId @param offerId @return OfferInquiryServiceResponse @throws FatalException	Fetch offer name from PMU @param sessionId @param creditCardAccountDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionId @param offerId @return OfferInquiryServiceResponse @throws FatalException	Validate new requested limit <ul> <li>Requested limit should be within the offer min / max limit</li> <li>Pre-Approved limit should not be null and requested limit should be less than or equal to pre-approved limit</li> <li>Requested limit should not be more than the current outstanding limit</li> <li>Requested limit should not be same as previous approved / sanctioned amount</li> </ul> @param sessionId @param requestLimit @param creditCardDetailsDTO
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionId @param offerId @return OfferInquiryServiceResponse @throws FatalException	Populate account facility details to Credit card details DTO @param sessionId @param creditCardDetailsDTO @param creditCardAccountDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	Fetch offer name from PMU @param sessionContext @param creditCardAccountDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardOfferReferences	This method inquires the credit card offer definition for the given offer id @param sessionContext @param offerId @return OfferInquiryServiceResponse @throws FatalException	<ol>This method would do call following functions internally to perform Individual Level Validations. <li>validatePrimaryCardHolderDetails</li> <li>validateBalanceTransferDetails</li> <li>validateDeliveryPreferencesDetails</li> <li>validateAddOnCardHolderDetails</li> <li>validateBillCodeDerivation</li> </ol> @param sessionContext @param accountConfigurationDTO
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardProductPreferences	This method inquires the credit card product preferences for the given product code @param sessionContext @param productCode @return CreditCardInquiryResponse @throws FatalException	
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchCreditCardProductPreferences	This method inquires the credit card product preferences for the given product code @param sessionContext @param productCode @return CreditCardInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	This service will populate the default fields for Credit Facility DTO from Credit Card Details DTO. @param sessionContext @param creditCardDetailsDTO @param creditFacilityDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	Fetch facility category from credit policy template @param sessionContext @param productId @param businessUnit @return facilityCategory @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	Populate Credit Card Details Post Inquiry @param sessionContext @param creditCardDetailsDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	Populate account facility details to Credit card details DTO @param sessionContext @param creditCardDetailsDTO @param creditCardAccountDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroup	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	The method fetchBillCode is used to fetch the bill code based on the derivation fact provided.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,BillCodeDerivationRequestDTO,billCodeDerivationRequestDTO, ,This attribute contains account id, credit card account configuration dto and bundle id. @return {@link CreditCardBillCodeResponse} This class contains the information related to credit card account details. @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	The method inquireCardAccountDetails is used to fetch credit card account details for the given card account number.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String,accountId, ,This attribute denotes the card account number. @return {@link CreditCardAccountDetailsResponse} This class contains the information related to credit card account details. @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	This method is used to create credit facility using the external credit card details @param sessionContext @param creditCardDetailsDTO @param borrowingEntityId @return CreditFacilityDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	This service will populate the default fields for Credit Facility DTO from Credit Card Details DTO. @param sessionContext @param creditCardDetailsDTO @param creditFacilityDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	Fetch facility category from credit policy template @param sessionContext @param productGroupId @param businessUnit @return facilityCategory @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	Populate Credit Card Details Post Inquiry @param sessionContext @param creditCardDetailsDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	Populate account facility details to Credit card details DTO @param sessionContext @param creditCardDetailsDTO @param creditCardAccountDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	This method inquires the product group for the product group code @param sessionContext @param productGroupCode @return ProductGroupDTO @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProduct	This represents the fetchProduct method of the entity. @param sessionContext @param productCode @return @throws FatalException	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	validateProductGroup	This method is used to validate product group @param sessionContext @param productGroup @return @throws FatalException	This method is used to validate product group @param sessionContext @param productGroup @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroupDetailsOnBusinessUnit	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException	Fetch facility category from credit policy template @param sessionContext @param productGroupId @param businessUnit @return facilityCategory @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroupDetailsOnBusinessUnit	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException	Populate Credit Card Details Post Inquiry @param sessionContext @param creditCardDetailsDTO @return @throws FatalException
ICardPMUAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPMUAdapter	fetchProductGroupDetailsOnBusinessUnit	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException	This method is used to fetch the product group details for the give product group id and business unit @param sessionContext @param productGroup @param businessUnit @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPMUAdpater	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdpater	fetchCreditPolicyDetails	This method is used to fetch credit policy template details for the given credit policy template id @param sessionContext @param creditPolicyTemplateId @return @throws FatalException	Fetch facility category from credit policy template @param sessionContext @param productId @param businessUnit @return facilityCategory @throws FatalException
ICardPMUAdpater	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdpater	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	Validate Add on card holder. This validation method does the below validations,  <ul> <li>Age of add on card holder not less than the minimum age maintained in the bank's policy parameter</li> <li>Add on card holder is allowed for this offer check</li> <li>Validate Maximum add-on cards for this offer</li> </ul> @param sessionContext @param addOnCardHolderDTOs @throws FatalException
ICardPMUAdpater	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPMUAdpater	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																														
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1" data-bbox="829 748 1276 1390"> <thead> <tr> <th data-bbox="829 748 1129 779">Module Type</th> <th data-bbox="1138 748 1276 779">Domain Category</th> </tr> </thead> <tbody> <tr> <td data-bbox="829 786 1129 816">null</td> <td data-bbox="1138 786 1276 816"></td> </tr> <tr> <td data-bbox="829 823 1129 854">ALL</td> <td data-bbox="1138 823 1276 854"></td> </tr> <tr> <td data-bbox="829 860 1129 891">UTILITY_PROVIDER</td> <td data-bbox="1138 860 1276 891"></td> </tr> <tr> <td data-bbox="829 898 1129 928">BILL_PAYMENTS</td> <td data-bbox="1138 898 1276 928"></td> </tr> <tr> <td data-bbox="829 935 1129 966">ATM</td> <td data-bbox="1138 935 1276 966"></td> </tr> <tr> <td data-bbox="829 972 1129 1003">ATM</td> <td data-bbox="1138 972 1276 1003"></td> </tr> <tr> <td data-bbox="829 1010 1129 1040">PAYMENTS</td> <td data-bbox="1138 1010 1276 1040"></td> </tr> <tr> <td data-bbox="829 1047 1129 1078">PAYMENTS</td> <td data-bbox="1138 1047 1276 1078"></td> </tr> <tr> <td data-bbox="829 1084 1129 1115">COLLECTION</td> <td data-bbox="1138 1084 1276 1115"></td> </tr> <tr> <td data-bbox="829 1122 1129 1153">COLLECTION</td> <td data-bbox="1138 1122 1276 1153"></td> </tr> <tr> <td data-bbox="829 1159 1129 1190">ORIGINATION</td> <td data-bbox="1138 1159 1276 1190"></td> </tr> <tr> <td data-bbox="829 1196 1129 1227">ORIGINATION</td> <td data-bbox="1138 1196 1276 1227"></td> </tr> <tr> <td data-bbox="829 1234 1129 1265">CUSTOMER_INFORMATION</td> <td data-bbox="1138 1234 1276 1265"></td> </tr> <tr> <td data-bbox="829 1271 1129 1302">PARTY</td> <td data-bbox="1138 1271 1276 1302"></td> </tr> </tbody> </table> <p>If the caller module is account based</p>	Module Type	Domain Category	null		ALL		UTILITY_PROVIDER		BILL_PAYMENTS		ATM		ATM		PAYMENTS		PAYMENTS		COLLECTION		COLLECTION		ORIGINATION		ORIGINATION		CUSTOMER_INFORMATION		PARTY		<p>The following method is called to return the settlement Amount tag. @param context @param preference @param productCode @param eventCode @param settlementPmuAdapter @param module @return @throws FatalException</p>
Module Type	Domain Category																																	
null																																		
ALL																																		
UTILITY_PROVIDER																																		
BILL_PAYMENTS																																		
ATM																																		
ATM																																		
PAYMENTS																																		
PAYMENTS																																		
COLLECTION																																		
COLLECTION																																		
ORIGINATION																																		
ORIGINATION																																		
CUSTOMER_INFORMATION																																		
PARTY																																		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1"> <thead> <tr> <th>Module Type</th> <th>Domain Category</th> </tr> </thead> <tbody> <tr> <td>ATM</td> <td>ATM</td> </tr> <tr> <td>ATM</td> <td>PAYMENTS</td> </tr> <tr> <td>PAYMENTS</td> <td>PAYMENTS</td> </tr> <tr> <td>COLLECTION</td> <td>COLLECTION</td> </tr> <tr> <td>COLLECTION</td> <td>ORIGINATION</td> </tr> <tr> <td>ORIGINATION</td> <td>ORIGINATION</td> </tr> <tr> <td>CUSTOMER_INFORMATION</td> <td>PARTY</td> </tr> </tbody> </table> <p>If the caller module is account based</p>	Module Type	Domain Category	ATM	ATM	ATM	PAYMENTS	PAYMENTS	PAYMENTS	COLLECTION	COLLECTION	COLLECTION	ORIGINATION	ORIGINATION	ORIGINATION	CUSTOMER_INFORMATION	PARTY	<p>This method is used to fetch the settlement amount tag. @param SessionContext sessionContext @param String productCode @param String eventCode @param Map &lt;String, String&gt; map @return String</p>
Module Type	Domain Category																			
ATM	ATM																			
ATM	PAYMENTS																			
PAYMENTS	PAYMENTS																			
COLLECTION	COLLECTION																			
COLLECTION	ORIGINATION																			
ORIGINATION	ORIGINATION																			
CUSTOMER_INFORMATION	PARTY																			

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc												
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1"> <thead> <tr> <th>Module Type</th> <th>Domain Category</th> </tr> </thead> <tbody> <tr> <td>ATM</td> <td>ATM</td> </tr> <tr> <td>PAYMENTS</td> <td>PAYMENTS</td> </tr> <tr> <td>COLLECTION</td> <td>COLLECTION</td> </tr> <tr> <td>ORIGINATION</td> <td>ORIGINATION</td> </tr> <tr> <td>CUSTOMER_INFORMATION</td> <td>PARTY</td> </tr> </tbody> </table> <p>If the caller module is account based</p>	Module Type	Domain Category	ATM	ATM	PAYMENTS	PAYMENTS	COLLECTION	COLLECTION	ORIGINATION	ORIGINATION	CUSTOMER_INFORMATION	PARTY	<p>This method is used to fetch settlement amount tag .</p> <p>@param sessionContext @param productCode @param eventCode @param callerModule @param direction @param service @param amountTag @return @throws FatalException</p>
Module Type	Domain Category															
ATM	ATM															
PAYMENTS	PAYMENTS															
COLLECTION	COLLECTION															
ORIGINATION	ORIGINATION															
CUSTOMER_INFORMATION	PARTY															

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																								
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1" data-bbox="829 755 1276 1388"> <thead> <tr> <th data-bbox="829 755 1129 786">Module Type</th> <th data-bbox="1138 755 1276 786">Domain Category</th> </tr> </thead> <tbody> <tr> <td data-bbox="829 792 1129 823">null</td> <td data-bbox="1138 792 1276 823">null</td> </tr> <tr> <td data-bbox="829 829 1129 860">ALL</td> <td data-bbox="1138 829 1276 860">UTILITY_PROVIDER</td> </tr> <tr> <td data-bbox="829 867 1129 898">ATM</td> <td data-bbox="1138 867 1276 898">BILL_PAYMENTS</td> </tr> <tr> <td data-bbox="829 904 1129 935">ATM</td> <td data-bbox="1138 904 1276 935">PAYMENTS</td> </tr> <tr> <td data-bbox="829 941 1129 974">ATM</td> <td data-bbox="1138 941 1276 974">PAYMENTS</td> </tr> <tr> <td data-bbox="829 980 1129 1011">ATM</td> <td data-bbox="1138 980 1276 1011">COLLECTION</td> </tr> <tr> <td data-bbox="829 1018 1129 1049">ATM</td> <td data-bbox="1138 1018 1276 1049">COLLECTION</td> </tr> <tr> <td data-bbox="829 1055 1129 1086">ATM</td> <td data-bbox="1138 1055 1276 1086">ORIGINATION</td> </tr> <tr> <td data-bbox="829 1092 1129 1123">ATM</td> <td data-bbox="1138 1092 1276 1123">ORIGINATION</td> </tr> <tr> <td data-bbox="829 1130 1129 1161">ATM</td> <td data-bbox="1138 1130 1276 1161">CUSTOMER_INFORMATION</td> </tr> <tr> <td data-bbox="829 1167 1129 1198">ATM</td> <td data-bbox="1138 1167 1276 1198">PARTY</td> </tr> </tbody> </table> <p>If the caller module is account based</p>	Module Type	Domain Category	null	null	ALL	UTILITY_PROVIDER	ATM	BILL_PAYMENTS	ATM	PAYMENTS	ATM	PAYMENTS	ATM	COLLECTION	ATM	COLLECTION	ATM	ORIGINATION	ATM	ORIGINATION	ATM	CUSTOMER_INFORMATION	ATM	PARTY	<p>This method is used to fetch settlement amount tag for domain event. @param sessionContext @param eventCode @param direction @param domainCategory @param amountTag @return @throws FatalException</p>
Module Type	Domain Category																											
null	null																											
ALL	UTILITY_PROVIDER																											
ATM	BILL_PAYMENTS																											
ATM	PAYMENTS																											
ATM	PAYMENTS																											
ATM	COLLECTION																											
ATM	COLLECTION																											
ATM	ORIGINATION																											
ATM	ORIGINATION																											
ATM	CUSTOMER_INFORMATION																											
ATM	PARTY																											

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																														
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1" data-bbox="835 755 1278 1388"> <thead> <tr> <th data-bbox="835 755 1129 812">Module Type</th> <th data-bbox="1129 755 1278 812">Domain Category</th> </tr> </thead> <tbody> <tr> <td data-bbox="835 812 1129 868">null</td> <td data-bbox="1129 812 1278 868"></td> </tr> <tr> <td data-bbox="835 868 1129 925">ALL</td> <td data-bbox="1129 868 1278 925"></td> </tr> <tr> <td data-bbox="835 925 1129 982">UTILITY_PROVIDER</td> <td data-bbox="1129 925 1278 982"></td> </tr> <tr> <td data-bbox="835 982 1129 1039">BILL_PAYMENTS</td> <td data-bbox="1129 982 1278 1039"></td> </tr> <tr> <td data-bbox="835 1039 1129 1096">ATM</td> <td data-bbox="1129 1039 1278 1096"></td> </tr> <tr> <td data-bbox="835 1096 1129 1153">ATM</td> <td data-bbox="1129 1096 1278 1153"></td> </tr> <tr> <td data-bbox="835 1153 1129 1209">PAYMENTS</td> <td data-bbox="1129 1153 1278 1209"></td> </tr> <tr> <td data-bbox="835 1209 1129 1266">PAYMENTS</td> <td data-bbox="1129 1209 1278 1266"></td> </tr> <tr> <td data-bbox="835 1266 1129 1323">COLLECTION</td> <td data-bbox="1129 1266 1278 1323"></td> </tr> <tr> <td data-bbox="835 1323 1129 1380">COLLECTION</td> <td data-bbox="1129 1323 1278 1380"></td> </tr> <tr> <td data-bbox="835 1380 1129 1437">ORIGINATION</td> <td data-bbox="1129 1380 1278 1437"></td> </tr> <tr> <td data-bbox="835 1437 1129 1494">ORIGINATION</td> <td data-bbox="1129 1437 1278 1494"></td> </tr> <tr> <td data-bbox="835 1494 1129 1550">CUSTOMER_INFORMATION</td> <td data-bbox="1129 1494 1278 1550"></td> </tr> <tr> <td data-bbox="835 1550 1129 1607">PARTY</td> <td data-bbox="1129 1550 1278 1607"></td> </tr> </tbody> </table> <p>&lt;/tbody&gt; &lt;/table&gt; &lt;/p&gt; If the caller module is account based</p>	Module Type	Domain Category	null		ALL		UTILITY_PROVIDER		BILL_PAYMENTS		ATM		ATM		PAYMENTS		PAYMENTS		COLLECTION		COLLECTION		ORIGINATION		ORIGINATION		CUSTOMER_INFORMATION		PARTY		
Module Type	Domain Category																																	
null																																		
ALL																																		
UTILITY_PROVIDER																																		
BILL_PAYMENTS																																		
ATM																																		
ATM																																		
PAYMENTS																																		
PAYMENTS																																		
COLLECTION																																		
COLLECTION																																		
ORIGINATION																																		
ORIGINATION																																		
CUSTOMER_INFORMATION																																		
PARTY																																		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc																								
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchSettlementAmountTag	<p>This method will inquire the settlement amount tag associated with the event.&lt;br&gt; An event can be domain category based or product based.&lt;br&gt; If the caller module is non-product module, then the caller can either send product code as null &lt;br&gt; OR&lt;br&gt; send both the isDomainEvent = true and the corresponding domain category.&lt;br&gt; In case of null product code, the method will infer the domain category based on the caller module type using the following chart</p> <table border="1" data-bbox="829 755 1283 1388"> <thead> <tr> <th data-bbox="829 755 1129 786">Module Type</th> <th data-bbox="1138 755 1283 786">Domain Category</th> </tr> </thead> <tbody> <tr> <td data-bbox="829 792 1129 823">null</td> <td data-bbox="1138 792 1283 823">null</td> </tr> <tr> <td data-bbox="829 829 1129 860">ALL</td> <td data-bbox="1138 829 1283 860">UTILITY_PROVIDER</td> </tr> <tr> <td data-bbox="829 867 1129 898">ATM</td> <td data-bbox="1138 867 1283 898">BILL_PAYMENTS</td> </tr> <tr> <td data-bbox="829 904 1129 935">ATM</td> <td data-bbox="1138 904 1283 935">PAYMENTS</td> </tr> <tr> <td data-bbox="829 941 1129 974">ATM</td> <td data-bbox="1138 941 1283 974">PAYMENTS</td> </tr> <tr> <td data-bbox="829 980 1129 1011">ATM</td> <td data-bbox="1138 980 1283 1011">COLLECTION</td> </tr> <tr> <td data-bbox="829 1018 1129 1049">ATM</td> <td data-bbox="1138 1018 1283 1049">COLLECTION</td> </tr> <tr> <td data-bbox="829 1055 1129 1086">ATM</td> <td data-bbox="1138 1055 1283 1086">ORIGINATION</td> </tr> <tr> <td data-bbox="829 1092 1129 1123">ATM</td> <td data-bbox="1138 1092 1283 1123">ORIGINATION</td> </tr> <tr> <td data-bbox="829 1130 1129 1161">ATM</td> <td data-bbox="1138 1130 1283 1161">CUSTOMER_INFORMATION</td> </tr> <tr> <td data-bbox="829 1167 1129 1198">ATM</td> <td data-bbox="1138 1167 1283 1198">PARTY</td> </tr> </tbody> </table> <p>If the caller module is account based</p>	Module Type	Domain Category	null	null	ALL	UTILITY_PROVIDER	ATM	BILL_PAYMENTS	ATM	PAYMENTS	ATM	PAYMENTS	ATM	COLLECTION	ATM	COLLECTION	ATM	ORIGINATION	ATM	ORIGINATION	ATM	CUSTOMER_INFORMATION	ATM	PARTY	<p>This method is called from the settlement taskflow.This method is used to collect all the settlement related information.</p>
Module Type	Domain Category																											
null	null																											
ALL	UTILITY_PROVIDER																											
ATM	BILL_PAYMENTS																											
ATM	PAYMENTS																											
ATM	PAYMENTS																											
ATM	COLLECTION																											
ATM	COLLECTION																											
ATM	ORIGINATION																											
ATM	ORIGINATION																											
ATM	CUSTOMER_INFORMATION																											
ATM	PARTY																											

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module, then the event is assumed to be product linked event. In case the event is domain based, and is being invoked from an account based module, then one must send isDomainEvent = true, and the corresponding domain category, along with the product code. &lt;br&gt; For ex, CASH Deposit on CASA is a domain event that is being invoked from Account based module.</p> <p>@param callerModule TODO  @param direction TODO @param direction Indicates the direction of fund flow from Customer perspective @param isDomainEvent TODO @param domainCategory TODO @param amountTag TODO @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal  @param amount The amount to be settled @param details The settlement mode details to be used @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException TODO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	resolveCashGl	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	resolveEndPointGl	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	resolveCardGl	<p>This is the basic method for invocation of module API based on the settlement mode. @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param amount The amount to be settled @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @throws FatalException</p>	
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	isModeCombinationValid		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		This represents the addComponentPreference method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,ComponentPreferenceDTO,componentPreferenceDTO, ,componentPreferenceDTO TODO: Explain the componentPreferenceDTO above @param eventCode @param productCodea @return {@link com.ofss.fc.service.response.TransactionStatus} @throws FatalException @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		<p>This method will validate that the settlement modes set up for setting up an instruction must match the settlement modes set up for the instruction execution. </p> @param sessionContext @param productCode @param originalEvent @param executionEvent @param modes @throws FatalException
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		This method is used to fetch allowed modes. @param SessionContext sessionContext @param String productCode @param String eventCode @param TaskModuleType callerModule @param SettlementModeType [] settlementModeType @return boolean
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		This method is used to fetch the allowed modes for settlement.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		This method is used to set the allowed modes for settlement.
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchAllowedModes		This method is used to create the amount component which is then stored in the component preference DTO. @param context @param amountComponentID @param moduleCode @param productCode @param eventCode @return @throws FatalException
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchExchangeRateType	This method returns the rate types to be used for each settlement mode for a given event code and either product or module. If product is supplied, then the module is not taken into consideration. The order in which the modes are passed is used to returning back the rate types for each mode @param context @param productCode @param eventCode @param callerModule @param modeTypes @return @throws FatalException	The following method does not perform any interaction @param sessionContext @param componentPreferenceDTO @param eventCode @param allowedSplitTypes @param productCode @param fromOfferId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchExchangeRateType	This method returns the rate types to be used for each settlement mode for a given event code and either product or module. If product is supplied, then the module is not taken into consideration. The order in which the modes are passed is used to returning back the rate types for each mode @param context @param productCode @param eventCode @param callerModule @param modeTypes @return @throws FatalException	
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchClearingBridgeLedger		This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction INDicates the direction of fund flow from Customer perspective @param transactionType INDicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchClearingBridgeLedger		This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context INdicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction Indicates the direction of fund flow from Customer perspective @param transactionType Indicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchClearingBridgeLedger		This method is used to fetch clearing Bridge Ledger @param SessionContext sessionContext @return String
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchClearingBridgeLedger		
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForProduct	<p>This method will return rate type for the input product code and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. </p> @param productCode @param eventId @param settlementModeType	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return @throws FatalException	
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForProduct	<p>This method will return rate type for the input product code and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. </p> @param productCode @param eventId @param settlementModeType @return @throws FatalException	<p>This method will fetch the rate type from PMU. This rate type will be either for productCode or nonProductDomainCategory. If settlementModeType is supplied, it iterates the response and fetches the rateType corresponding to the input settlementModeType, else it fetches the record will event code as AL. </p> @param productCode @param eventId @param nonProductDomainCategory @param settlementModeType @return
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForProduct	<p>This method will return rate type for the input product code and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. </p> @param productCode @param eventId @param settlementModeType @return @throws FatalException	
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForNonProduct	<p>This method will return rate type for the input non product domain category and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. </p> @param nonProductDomainCategory @param eventId @param settlementModeType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForNonProduct	<p>&lt;p&gt;This method will return rate type for the input non product domain category and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. &lt;/p&gt;           @param nonProductDomainCategory @param eventId @param settlementModeType @return @throws FatalException</p>	<p>&lt;p&gt;This method will fetch the rate type from PMU. This rate type will be either for productCode or nonProductDomainCategory. If settlementModeType is supplied, it iterates the response and fetches the rateType corresponding to the input settlementModeType, else it fetches the record will event code as AL. &lt;/p&gt;           @param productCode @param eventId @param nonProductDomainCategory @param settlementModeType @return</p>
ISettlementPmuAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementPmuAdapter	fetchRateTypeForNonProduct	<p>&lt;p&gt;This method will return rate type for the input non product domain category and event id that corresponds to the input settlement mode type. If no rate matches the input settlement mode type, it returns the one with event code 'AL'. &lt;/p&gt;           @param nonProductDomainCategory @param eventId @param settlementModeType @return @throws FatalException</p>	
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	transferFundToBeneficiary	<p>@param casaToCasaRequestDTO @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	transferFundToBeneficiary	@param casaToCasaRequestDTO @return @throws FatalException	The following method is used to execute casa account related transaction. @param businessContext specifies party and account related information. @param accountType specifies account type @param namedPartyRelationshipDTO specifies namedPartyRelationshipDTO @param fromAccount specifies the account which is being debited @param toAccount specifies the account which is being credited @param amount specifies the amount of the transaction. @param depositNo specifies the deposit no if it is a TD account. @param isInflowTransaction specifies the direction of the transaction. @param chequeNo specifies the cheque no. @param date specifies the date. @param term specifies the term. @param primaryReasonCode specifies the primary reason code. @param secondaryReasonCode specifies the secondary reason code. @param purpose specifies the purpose. @param contextResponse specifies the response class. @throws FatalException
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	setupSweepInRequest	@param sweepInRequestDTO @return @throws FatalException	
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	setupSweepInRequest	@param sweepInRequestDTO @return @throws FatalException	The following method is used to setup sweepin instruction command . It extracts various information from the command which is necessary to setup the sweepin instruction. @author AnukoolA

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	setUpSweepOutRequest	@param sweepOutRequestDTO @return @throws FatalException	The following method is used to setup sweep out to a deposit command . It extracts various information from the command which is necessary to setup the sweep out instruction to a deposit..
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	setUpSweepOutRequest	@param sweepOutRequestDTO @return @throws FatalException	The following method is being used to create the sweep out instruction. @param contextResponse @param businessContext @param accountType @param taskSetUpSweepOutToAccount @throws FatalException
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	setUpSweepOutRequest	@param sweepOutRequestDTO @return @throws FatalException	
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doCashDeposit	@param cashTransactionRequestDTO @return @throws FatalException	
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doCashDeposit	@param cashTransactionRequestDTO @return @throws FatalException	The following method is used to execute deposit cash command. Denomination is required only when denomination tracking is enabled.
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doCashWithdrawal	@param cashTransactionRequestDTO @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doCashWithdrawal	@param cashTransactionRequestDTO @return @throws FatalException	The following method is used to execute cash withdrawal for DDA account. @param contextResponse @param businessContext @param taskWithdraw @throws FatalException
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doStandingInstructionSetup	@param siRequestDTO @return @throws FatalException	The following method is used to setup standing instruction command . It extracts various information from the command which is necessary to setup the standing instruction. @author Anukoola
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doStandingInstructionSetup	@param siRequestDTO @return @throws FatalException	The following method is used to execute SI credit. @param contextResponse @param businessContext @param taskSetUpCreditSI @param demandDepositAdapter @param namedPartyRelationshipDTO @throws FatalException
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doStandingInstructionSetup	@param siRequestDTO @return @throws FatalException	
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doBillPaymentOneTime	@param billPaymentRequestDTO @return @throws FatalException	
IDSLEDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLEDemandDepositAdapter	doBillPaymentOneTime	@param billPaymentRequestDTO @return @throws FatalException	The following method is used to perform execution of transfer to biller commands. @param businessContextDTO @param accountType @param dashboardService @param taskTransferFromAccount @param namedPartyRelationshipDTO @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createChequeInstruction	<p>TODO: Check which service needs to be called for service charge and incorporate. This method is used to create Cheque Instructions. &lt;br&gt; The algorithm of the method is as follows : &lt;ul&gt; &lt;li&gt;Accept a DTO object ChequeInstructionDTO as input parameter.&lt;/li&gt; &lt;li&gt;Call the assembler {&lt;a href="#"&gt;@link ChequeInstructionAssembler#toDomainObject&lt;/a&gt;} to convert it into the ChequeInstruction Domain object.&lt;/li&gt; &lt;li&gt;The ChequeInstructionDTO can be an instance of any of the Instruction type, i.e. Stop Payment, Revoke Stop Payment or Stop Payment For Amount Range. The assembler will convert it into the relevant class and return. &lt;li&gt;Call the domain service {&lt;a href="#"&gt;@link ChequeInstructionDomainService#createChequeInstruction&lt;/a&gt;} which will create the instruction and also process its execution.&lt;/li&gt; &lt;/ul&gt;</p> <p>&lt;Note&gt; The caller of this method has to send the proper instance of the DTO, i.e., if the instruction type = Stop Payment, an instance of StopPaymentChequeInstructionDTO should be sent. @param chequeInstructionDTO @throws FatalException</p>	<p>The following method is used to stop cheque series. @param chequeStartNo @param chequeEndNo @param responseText @param chequeInstructionDTO @param stopPaymentDTO @param demandDepositAdapter @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createChequeInstruction	<p>TODO: Check which service needs to be called for service charge and incorporate. This method is used to create Cheque Instructions. &lt;br&gt; The algorithm of the method is as follows : &lt;ul&gt; &lt;li&gt;Accept a DTO object ChequeInstructionDTO as input parameter.&lt;/li&gt; &lt;li&gt;Call the assembler {&lt;a href="#"&gt;@link ChequeInstructionAssembler#toDomainObject&lt;/a&gt;} to convert it into the ChequeInstruction Domain object.&lt;/li&gt; &lt;li&gt;The ChequeInstructionDTO can be an instance of any of the Instruction type, i.e. Stop Payment, Revoke Stop Payment or Stop Payment For Amount Range. The assembler will convert it into the relevant class and return. &lt;li&gt;Call the domain service {&lt;a href="#"&gt;@link ChequeInstructionDomainService#createChequeInstruction&lt;/a&gt;} which will create the instruction and also process its execution.&lt;/li&gt; &lt;/ul&gt;</p> <p>&lt;Note&gt; The caller of this method has to send the proper instance of the DTO, i.e., if the instruction type = Stop Payment, an instance of StopPaymentChequeInstructionDTO should be sent. @param chequeInstructionDTO @throws FatalException</p>	<p>The following method is used to stop single leaf cheque. @param chequeStartNo @param responseText @param chequeInstructionDTO @param stopPaymentDTO @param demandDepositAdapter @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createChequeInstruction	<p>TODO: Check which service needs to be called for service charge and incorporate. This method is used to create Cheque Instructions.</p> <p>The algorithm of the method is as follows :</p> <ul style="list-style-type: none"> <li>Accept a DTO object ChequeInstructionDTO as input parameter.</li> <li>Call the assembler <code>{@link ChequeInstructionAssembler#toDomainObject}</code> to convert it into the ChequeInstruction Domain object.</li> <li>The ChequeInstructionDTO can be an instance of any of the Instruction type, i.e. Stop Payment, Revoke Stop Payment or Stop Payment For Amount Range. The assembler will convert it into the relevant class and return.</li> <li>Call the domain service <code>{@link ChequeInstructionDomainService#createChequeInstruction}</code> which will create the instruction and also process its execution.</li> </ul> <p><b>&lt;Note&gt;</b> The caller of this method has to send the proper instance of the DTO, i.e., if the instruction type = Stop Payment, an instance of StopPaymentChequeInstructionDTO should be sent. @param chequeInstructionDTO @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	fetchBalanceDetails	<p>Demand deposit account has various balances. This method is used to inquire balance information. This method provides following balances. &lt;ol&gt; &lt;li&gt;Clear balance &lt;li&gt;Uncleared Balance &lt;li&gt;Debit balances &lt;li&gt;Overdraft usage &lt;li&gt;Overline usage &lt;li&gt;TOD usage &lt;/ol&gt; &lt;li&gt;Uncleared balance usage: Advance taken against unclear funds. &lt;li&gt;Sweep in details : Funds which are taken from term deposit account. &lt;li&gt;Hold Amounts: system initiated or user initiated hold amount.it will include earmark. &lt;li&gt;Net Balance : &lt;/ol&gt; Some of the important points about this method mentioned below. &lt;ol&gt; &lt;li&gt;All balances are expressed in account currency only. &lt;li&gt;Except {@code netBalance}, All balances can not be negative. &lt;/ol&gt; &lt;br&gt;This method performs following steps &lt;ol&gt; &lt;li&gt;Get IDemandDepositAccount: Call DemandDepositAccountService.fetchDemandDepositAccount &lt;li&gt;Prepare transaction dates . &lt;li&gt;Call Appropriate methods of {@link IDemandDepositAccount} to get the balances. &lt;li&gt;Call {@code calculateNetbalance} to get net balance of the account. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, ,</p>	The following method is used to fetch the balances for the account which is in context. Ballance display pattern is different for different account type's.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId,&lt;br&gt; &lt;br&gt; The attribute required to fetch basic details of DDAccount.</p> <p>@return {@link BalanceInquiryDetailsResponse}&lt;br&gt; &lt;br&gt; BalanceInquiryDetailsResponse contains BalanceInquiryDetailsDTO which will contain following information.&lt;br&gt;</p> <p>&lt;b&gt;clearBalance&lt;/b&gt;Clear balance is customer's own funds in the account excluding any limits given to the customer/account .&lt;br&gt;</p> <p>&lt;b&gt;unclearedFunds&lt;/b&gt;This attribute denotes unclear balance of the account.&lt;br&gt;</p> <p>&lt;b&gt;utilizedOverdraftLimit&lt;/b&gt;This attribute denotes utilized overdraft limit.&lt;br&gt;</p> <p>&lt;b&gt;overlineAmount&lt;/b&gt;This attribute denotes utilized overline amount.&lt;br&gt;</p> <p>&lt;b&gt;todAmount&lt;/b&gt;This attribute denotes tod amount.&lt;br&gt;</p> <p>&lt;b&gt;tdSweepInLienAmount&lt;/b&gt;This attribute denotes sweep in lien amount. &lt;br&gt;</p> <p>&lt;b&gt;utilizedAUFLimit&lt;/b&gt; This attribute denotes utilised AUF</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>amount.&lt;br&gt;  &lt;b&gt;userInitiatedHoldAmount&lt;/b&gt;  This attribute denotes user initiated hold amount.&lt;br&gt;  &lt;b&gt;systemInitiatedHoldAmount&lt;/b&gt;  This attribute denotes system initiated hold amount.&lt;br&gt;  &lt;b&gt;netBalance&lt;/b&gt; This attributes denotes net balance of demand deposit account.&lt;br&gt;  &lt;b&gt;applicableAUFLimit&lt;/b&gt; This field denotes the amount available as a part of the advance against unclear funds.&lt;br&gt;  &lt;b&gt;totalHoldAmount&lt;/b&gt;This attribute total hold amount on demand deposit account.&lt;br&gt;  &lt;b&gt;odLimit&lt;/b&gt;This attribute denotes total overdraft limit present .&lt;br&gt; In case of failure -&lt;br&gt;  &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt;</li></ul> @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	fetchBalanceDetails	<p>Demand deposit account has various balances. This method is used to inquire balance information. This method provides following balances. <ol style="list-style-type: none"> <li>Clear balance</li> <li>Uncleared Balance</li> <li>Debit balances</li> <li>Overdraft usage</li> <li>Overline usage</li> <li>TOD usage</li> </ol> <li>Uncleared balance usage: Advance taken against unclear funds. <li>Sweep in details : Funds which are taken from term deposit account. <li>Hold Amounts: system initiated or user initiated hold amount. it will include earmark. <li>Net Balance : </li>  Some of the important points about this method mentioned below. <ol style="list-style-type: none"> <li>All balances are expressed in account currency only. <li>Except {@code netBalance}, All balances can not be negative. </li> </li></ol>   This method performs following steps <ol style="list-style-type: none"> <li>Get IDemandDepositAccount: Call DemandDepositAccountService.fetchDemandDepositAccount</li> <li>Prepare transaction dates .</li> <li>Call Appropriate methods of {@link IDemandDepositAccount} to get the balances. <li>Call {@code calculateNetbalance} to get net balance of the account. </li> </li></ol> @fcb.param MI,SessionContext,sessionContext, ,</li></li></li></p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId,&lt;br&gt; &lt;br&gt; The attribute required to fetch basic details of DDAccount.</p> <p>@return {@link BalanceInquiryDetailsResponse}&lt;br&gt; &lt;br&gt; BalanceInquiryDetailsResponse contains BalanceInquiryDetailsDTO which will contain following information.&lt;br&gt;</p> <p>&lt;b&gt;clearBalance&lt;/b&gt;Clear balance is customer's own funds in the account excluding any limits given to the customer/account .&lt;br&gt;</p> <p>&lt;b&gt;unclearedFunds&lt;/b&gt;This attribute denotes unclear balance of the account.&lt;br&gt;</p> <p>&lt;b&gt;utilizedOverdraftLimit&lt;/b&gt;This attribute denotes utilized overdraft limit.&lt;br&gt;</p> <p>&lt;b&gt;overlineAmount&lt;/b&gt;This attribute denotes utilized overline amount.&lt;br&gt;</p> <p>&lt;b&gt;todAmount&lt;/b&gt;This attribute denotes tod amount.&lt;br&gt;</p> <p>&lt;b&gt;tdSweepInLienAmount&lt;/b&gt;This attribute denotes sweep in lien amount. &lt;br&gt;</p> <p>&lt;b&gt;utilizedAUFLimit&lt;/b&gt; This attribute denotes utilised AUF</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>amount.&lt;br&gt;  &lt;b&gt;userInitiatedHoldAmount&lt;/b&gt;  This attribute denotes user initiated hold amount.&lt;br&gt;  &lt;b&gt;systemInitiatedHoldAmount&lt;/b&gt;  This attribute denotes system initiated hold amount.&lt;br&gt;  &lt;b&gt;netBalance&lt;/b&gt; This attributes denotes net balance of demand deposit account.&lt;br&gt;  &lt;b&gt;applicableAUFLimit&lt;/b&gt; This field denotes the amount available as a part of the advance against unclear funds.&lt;br&gt;  &lt;b&gt;totalHoldAmount&lt;/b&gt;This attribute total hold amount on demand deposit account.&lt;br&gt;  &lt;b&gt;odLimit&lt;/b&gt;This attribute denotes total overdraft limit present .&lt;br&gt; In case of failure -&lt;br&gt;  &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt;</li></ul> @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	fetchBalanceDetails	<p>Demand deposit account has various balances. This method is used to inquire balance information. This method provides following balances. &lt;ol&gt; &lt;li&gt;Clear balance &lt;li&gt;Uncleared Balance &lt;li&gt;Debit balances &lt;li&gt;Overdraft usage &lt;li&gt;Overline usage &lt;li&gt;TOD usage &lt;/ol&gt; &lt;li&gt;Uncleared balance usage: Advance taken against unclear funds. &lt;li&gt;Sweep in details : Funds which are taken from term deposit account. &lt;li&gt;Hold Amounts: system initiated or user initiated hold amount. it will include earmark. &lt;li&gt;Net Balance : &lt;/ol&gt; Some of the important points about this method mentioned below. &lt;ol&gt; &lt;li&gt;All balances are expressed in account currency only. &lt;li&gt;Except {@code netBalance}, All balances can not be negative. &lt;/ol&gt; &lt;br&gt; This method performs following steps &lt;ol&gt; &lt;li&gt;Get IDemandDepositAccount: Call DemandDepositAccountService.fetchDemandDepositAccount &lt;li&gt;Prepare transaction dates . &lt;li&gt;Call Appropriate methods of {@link IDemandDepositAccount} to get the balances. &lt;li&gt;Call {@code calculateNetbalance} to get net balance of the account. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, ,</p>	<p>The following method is used to execute casa account related transaction. @param businessContext specifies party and account related information. @param accountType specifies account type @param namedPartyRelationshipDTO specifies namedPartyRelationshipDTO @param fromAccount specifies the account which is being debited @param toAccount specifies the account which is being credited @param amount specifies the amount of the transaction. @param depositNo specifies the deposit no if it is a TD account. @param isInflowTransaction specifies the direction of the transaction. @param chequeNo specifies the cheque no. @param date specifies the date. @param term specifies the term. @param primaryReasonCode specifies the primary reason code. @param secondaryReasonCode specifies the secondary reason code. @param purpose specifies the purpose. @param contextResponse specifies the response class. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,accountId,&lt;br&gt; &lt;br&gt; The attribute required to fetch basic details of DDAccount.</p> <p>@return {@link BalanceInquiryDetailsResponse}&lt;br&gt; &lt;br&gt; BalanceInquiryDetailsResponse contains BalanceInquiryDetailsDTO which will contain following information.&lt;br&gt;</p> <p>&lt;b&gt;clearBalance&lt;/b&gt;Clear balance is customer's own funds in the account excluding any limits given to the customer/account .&lt;br&gt;</p> <p>&lt;b&gt;unclearedFunds&lt;/b&gt;This attribute denotes unclear balance of the account.&lt;br&gt;</p> <p>&lt;b&gt;utilizedOverdraftLimit&lt;/b&gt;This attribute denotes utilized overdraft limit.&lt;br&gt;</p> <p>&lt;b&gt;overlineAmount&lt;/b&gt;This attribute denotes utilized overline amount.&lt;br&gt;</p> <p>&lt;b&gt;todAmount&lt;/b&gt;This attribute denotes tod amount.&lt;br&gt;</p> <p>&lt;b&gt;tdSweepInLienAmount&lt;/b&gt;This attribute denotes sweep in lien amount. &lt;br&gt;</p> <p>&lt;b&gt;utilizedAUFLimit&lt;/b&gt; This attribute denotes utilised AUF</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>amount.&lt;br&gt;  &lt;b&gt;userInitiatedHoldAmount&lt;/b&gt;  This attribute denotes user initiated hold amount.&lt;br&gt;  &lt;b&gt;systemInitiatedHoldAmount&lt;/b&gt;  This attribute denotes system initiated hold amount.&lt;br&gt;  &lt;b&gt;netBalance&lt;/b&gt; This attributes denotes net balance of demand deposit account.&lt;br&gt;  &lt;b&gt;applicableAUFLimit&lt;/b&gt; This field denotes the amount available as a part of the advance against unclear funds.&lt;br&gt;  &lt;b&gt;totalHoldAmount&lt;/b&gt;This attribute total hold amount on demand deposit account.&lt;br&gt;  &lt;b&gt;odLimit&lt;/b&gt;This attribute denotes total overdraft limit present .&lt;br&gt; In case of failure -&lt;br&gt;  &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; <ul style="list-style-type: none"> <li>&lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; </li></ul> @throws FatalException</p>	
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createChequeBookRequest	<p>The following method is used to raise a cheque book request.  @param inputRequestDTO @return @throws FatalException</p>	<p>The following method is used to request cheque book for a DDA account. @param contextResponse @param businessContext @param demandDepositAdapter @param requestChequeBook @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createChequeBookRequest	The following method is used to raise a cheque book request. @param inputRequestDTO @return @throws FatalException	
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	procureFundsFromExternalAccount	The following method is used to procure funds from external account. @param sessionContext @param initiatorPartyId @param initiatorAccountId @param counterpartyId @param description @param transactionAmount @return @throws FatalException	
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	procureFundsFromExternalAccount	The following method is used to procure funds from external account. @param sessionContext @param initiatorPartyId @param initiatorAccountId @param counterpartyId @param description @param transactionAmount @return @throws FatalException	The following method is used to execute casa account related transaction. @param businessContext specifies party and account related information. @param accountType specifies account type @param namedPartyRelationshipDTO specifies namedPartyRelationshipDTO @param fromAccount specifies the account which is being debited @param toAccount specifies the account which is being credited @param amount specifies the amount of the transaction. @param depositNo specifies the deposit no if it is a TD account. @param isInflowTransaction specifies the direction of the transaction. @param chequeNo specifies the cheque no. @param date specifies the date. @param term specifies the term. @param primaryReasonCode specifies the primary reason code. @param secondaryReasonCode specifies the secondary reason code. @param purpose specifies the purpose. @param contextResponse specifies the response class. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	fetchChequeBookIssuance	<p>The method will be used for inquiry of issued chequebooks. The algorithm of the method is as follows : &lt;ul&gt; &lt;li&gt;It accepts the ChChequeBookIssuanceDTO as an input parameter.&lt;/li&gt; &lt;li&gt;A call is made to the assembler {@link ChequeManagementAssembler#toIssuanceObject} to convert it into the ChequeBookIssuance domain object.&lt;/li&gt; &lt;li&gt;Finally the object is passed to the domain service {@link ChequeManagementService#fetchChequeBookIssuance} which will fetch the record from the database.&lt;/li&gt; &lt;li&gt;The assembler method {@link ChequeManagementService#fromIssuanceObject} is called to convert it into ChChequeBookIssuanceDTO&lt;/li&gt;&lt;/ul&gt; @param issuanceDTO</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createClosureRequest	<p>TODO: The request needs to be sent to the BEPL worklist. Need to check why it is not allowing 2 out parameters. This method allows creation of an account closure request. An account closure request simply marks the account for closure and does not actually close the account. The account closure request is subsequently evaluated by a relationship manager who eventually decides whether to go ahead with the actual closure or retain the account after talking to the customer. The method works as follows :</p> <ul style="list-style-type: none"> <li>Accept the ClosureRequestDTO as the input parameter.</li> <li>Call the assembler method <a href="#">{@link DDAClosureAssembler#toDomainObject}</a> to convert the DTO into the domain object</li> <li>Call the domain service method <a href="#">{@link DDAClosureService#createClosureRequest}</a> to post the request and mark the account for closure</li> </ul> <p>@param closureRequestDTO @throws FatalException</p>	
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	activateAccount	<p>The following method is used to activate account status from Dormant to regular.</p> <p>@param sessionContext @param accountId @return @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createASPINstruction		The following method is used to execute SI credit for a casa account. @param contextResponse @param businessContext @param taskSetUpCreditSI @param stringBuilder @param demandDepositAdapter @param identifier @throws FatalException
IDSLDemandDepositAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLDemandDepositAdapter	createASPINstruction		
IServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\commons\service\IServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method takes in the verified checklist ID and returns the   {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.    For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext : {@link SessionContext} @param verificationChecklistId : {@link String} @return : {@link VerificationChecklistResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will take a checklist and return an arraylist contain   verifications and associated verification entity linkage belong to that   checklist @param checklist : {@link VerificationChecklist} @param sessionContext : {@link SessionContext} @return : ArrayList of {@link VerificationDTO} @throws Throwable : If any error occurs, {@link Throwable} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will return an array of {@link:VerificationDocTypeLinkageDTO}   belong to a given a VerificationTypeDTO @param verificationTypeDTO : {@link VerificationTypeDTO} @return : Array of {@link VerificationDocTypeLinkageDTO}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeId and categoryId @param sessionId @param ruleId @param facts [] @return typeCategoryDTO	This method creates a new Verification Checklist in the system. A list of verifications is   required as an input. createChecklist will build the checklist and linkages from the   input and persist the data to the database. It will also return the   {@link CreateVerificationChecklistResponse} which contains the checklist ID to   This method will call createChecklistGroup to create checklist group before adding the checklist and then calling addChecklistToGroup to add   checklist in respective group. the caller.    For example, submitting new created application in Origination will trigger an event to   build the verification checklist. Origination will then pass this checklist to   VerificationApplicationService to persist in to the Database. createChecklist will return   the verification checklist ID so that Origination can store this value for inquiry. @param sessionId : {@link SessionContext} @param checklistDTO : {@link VerificationChecklistDTO} @return : {@link VerificationChecklistGroupCreationResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method creates a new Verification Checklist in the system. A list of verifications is   required as an input. addChecklistToGroup will build the checklist and linkages from   the input and persist the data to the database. It will also return the   {@link CreateVerificationChecklistResponse} which contains the checklist ID to the   caller.  In addition, the method also requires the check list group ID to link the verification.    For example, submitting new created application in Origination will trigger an event to   build the verification checklist. Origination will then pass this checklist to   VerificationApplicationService to persist in to the Database.   VerificationApplicationService also link the new checklist to the existing group.   addVerificationChecklistToGroup will also return the verification checklist ID so that   Origination can store this value for inquiry. @param sessionContext : {@link SessionContext} @param checklist : {@link VerificationChecklistDTO} @param verificationChecklistId : {@link String} @return : {@link VerificationChecklistCreationResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This methods updates the verification checklist. All the existing verifications in the   verification checklist will be updated. The updated information includes user comments,   linked documents and new verification added by user. All the data will be updated in   one function call. The method requires the checklist ID as an input. It will load the   checklist and update all the verifications and linkages in the checklist.    For example, a banker adds some uploaded documents to some verification in a   checklist. He or she also add some comments and an ad-hoc verification. All the   information will be captured in the checklist and send to the VerificationApplicationService. @param sessionContext : {@link SessionContext} @param checklist : {@link VerificationChecklistDTO} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This methods updates the verification checklist. All the existing verifications in the   verification checklist will be updated. The updated information includes user comments,   linked documents and new verification added by user. All the data will be updated in   one function call. The method requires the checklist ID as an input. It will load the   checklist and update all the verifications and linkages in the checklist.    For example, a banker adds some uploaded documents to some verification in a   checklist. He or she also add some comments and an ad-hoc verification. All the   information will be captured in the checklist and send to the VerificationApplicationService. @param checklist : {@link VerificationChecklistDTO} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeId and categoryId @param sessionId @param ruleId @param facts [] @return typeCategoryDTO	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param sessionId : {@link SessionContext} @param verificationContext : {@link VerificationContextDTO} @param effectiveDate : {@link Date} @param entityLinkageLevelType : {@link EntityLinkageLevelType} @param facts : Array of {@link FactInformationDTO} @return : {@link VerificationTypeResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeId and categoryId @param sessionId @param ruleId @param facts [] @return typeCategoryDTO	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param sessionId : {@link SessionContext} @param verificationContext : {@link VerificationContextDTO} @param effectiveDate : {@link Date} @param entityLinkageLevelType : {@link EntityLinkageLevelType} @param facts : Array of {@link FactInformationDTO} @param ruleId : {@link String} @return : {@link VerificationTypeResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will take an array of type Ids and in turn,   query {@link VerificationTypeService} to return an array of   VerificationTypeDTOs @param sessionContext : {@link SessionContext} @param typeIds : Array of {@link String} @return : Array of {@link VerificationTypeDTO} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will take a single type Id and in turn,   query {@link VerificationTypeService} to return a   VerificationTypeDTO @param sessionContext : {@link SessionContext} @param typeId : {@link String} @return : {@link VerificationTypeDTO} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return	This method will take an array of category Ids and in turn,   query {@link VerificationCategoryService} to return an array of   VerificationCategoryDTOs @param sessionContext : {@link SessionContext} @param categoryIds : Array of {@link String} @return : Array of {@link VerificationCategoryDTO} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			typeCategoryDTO	
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will take a category Id and in turn,   query {@link:VerificationCategoryService} to return a   VericationCategoryDTO @param sessionContext : {@link SessionContext} @param categoryId : {@link String} @return : {@link VerificationCategoryDTO} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\commons-service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This service will be called whenever history data for a particular checklistId needs to be created. The verifications, linkages and document linkages will be moved to the respective history tables and will be deleted from the original tables. @param sessionContext : {@link SessionContext} @param verificationChecklistId : {@link String} @return : {@link TransactionStatus} @throws FatalException : If any error occurs, {@link FatalException} is thrown.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method takes in the verified checklist ID and returns the   {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.    For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. This method will additionally return the history of the verifications if any fetched on the basis of the auxiliary linkages sent in the verificationHistoryInquiryContextDTO @param sessionContext : {@link SessionContext} @param verificationChecklistId : {@link String} @param verificationHistoryInquiryContextDTO : {@link VerificationHistoryInquiryContextDTO} @return : {@link VerificationChecklistResponse} @throws FatalException : If any error occurs, {@link FatalException} is thrown.
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commons\service\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICommonServiceRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\commonservice\ICommonServiceRuleAdapter	getVerification	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO	This method will query Rule engine to get a list of   type and category ids for expected Verifications. This method expect a rule id to locate the rule it will query   and a list of facts to establish the rule context. This function return a DTO contain 2 string arrays:   typeIds and categoryIds @param sessionContext @param ruleId @param facts [] @return typeCategoryDTO
IDSLCommonServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLCommonServiceAdapter	addComment		
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logDDAAction	This method will fetch the CASA account and then will get the party id and branch id and then log the DDAAction using action logger @param accountId @param actionDueDate	This method will be called after account or deposit is opened for adding rate variances to the pricing plan after account/deposit is opened i.e with arrangement number/account number @param sessionContext : holds the basic information about the user of the system @param arrangementNumber : the arrangement number of the party @param domain : the domain name like Banking, Insurance @param domainCategory : the domain category name like CASA, TD @param eventRateVarianceDTOs : array of EventRateVarianceDTO which the encapsulates the rate variance information for a particular event @return TransactionStatus : holds information on the transaction status @throws FatalException if the pricing contract service throws a FatalException @author: SureshRagala

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logDDAAction	This method will fetch the CASA account and then will get the party id and branch id and then log the DDAAction using action logger @param accountId @param actionDueDate	This method will be called after account or deposit is opened for adding rate variances to the pricing plan after account/deposit is opened i.e with arrangement number/account number.  This should be used <b><em>while UDM addition is not done by ELPBC screen</em></b>. @param sessionContext : holds the basic information about the user of the system @param arrangementNumber : the arrangement number of the party @param domain : the domain name like Banking, Insurance @param domainCategory : the domain category name like CASA, TD @param eventRateVarianceDTOs : array of EventRateVarianceDTO which the encapsulates the rate variance information for a particular event @return TransactionStatus : holds information on the transaction status @throws FatalException if the pricing contract service throws a FatalException @author: SudipG
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logDDAAction	This method will fetch the CASA account and then will get the party id and branch id and then log the DDAAction using action logger @param accountId @param actionDueDate	
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logDDAAction	This method will fetch the CASA account and then will get the party id and branch id and then log the DDAAction using action logger @param accountId @param actionDueDate	This method will call the pricing account adapter {@link IPricingAccountAdapter} and log the DDA action for CASA accounts @param accountNo @param repriceDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logDDAAction	This method will fetch the CASA account and then will get the party id and branch id and then log the DDAAction using action logger @param accountId @param actionDueDate	This method will save the data in to Repricing Table namely {@link RepriceStageData} and {@link RepriceAccountStageData} whenever there is a change in UDM. Also if the account in CASA then it will logDDA Action for current and future dated UDM changes or will logBackDatedAction for backDated UDM change @param eventRateVariances @param arrangementNumber @param domainCategory @throws FatalException
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getBasicDetailsForAccount	This method will return the basic details of the given account number and domain category. It will return a {@link String} Array with 3 elements:  Element at index 0: the Branch Code  Element at index 1: the Party Id  Element at index 2: the Account Currency Code  @param accountId the Account Id @param domainCategory the Domain Category Id @return the basic details of Account @throws FatalException	To get the branch code for the given arrangement number and domain category @param arrangementNumber @param domainCategory @return branchCode
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getBasicDetailsForAccount	This method will return the basic details of the given account number and domain category. It will return a {@link String} Array with 3 elements:  Element at index 0: the Branch Code  Element at index 1: the Party Id  Element at index 2: the Account Currency Code  @param accountId the Account Id @param domainCategory the Domain Category Id @return the basic	To get the currency code for the given arrangement number and domain category @param factContext @param arrangementNumber @param domainCategory @return currencyCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			details of Account @throws FatalException	
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getBasicDetailsForAccount	This method will return the basic details of the given account number and domain category. It will return a {@link String} Array with 3 elements:  Element at index 0: the Branch Code  Element at index 1: the Party Id  Element at index 2: the Account Currency Code  @param accountId the Account Id @param domainCategory the Domain Category Id @return the basic details of Account @throws FatalException	To get the branch code for the given arrangement number and domain category @param arrangementNumber @param domainCategory @return branchCode
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getBasicDetailsForAccount	This method will return the basic details of the given account number and domain category. It will return a {@link String} Array with 3 elements:  Element at index 0: the Branch Code  Element at index 1: the Party Id  Element at index 2: the Account Currency Code  @param accountId the Account Id @param domainCategory the Domain Category Id @return the basic details of Account @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	@see com.ofss.fc.framework.batch.service.action.IActionExecutor#executeAction(com.ofss.fc.framework.context.ApplicationContext, com.ofss.fc.framework.batch.data.ActionKey, com.ofss.fc.framework.domain.entity.action.ActionRequest, com.ofss.fc.datatype.Date)
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	This method processes the given deferred re-price action. It calls the respective adapter (depending on the given domain category code) to fetch the re-price configuration information {@link RepriceFormDTO}.   If the {@link RepriceProcessingAction} of the re-price form is {@link RepriceProcessingAction#ACCEPT}, then the pricing plan is re-priced for the applicable pricing element.   If the {@link RepriceProcessingAction} of the re-price form is {@link RepriceProcessingAction#REJECT} or {@link RepriceProcessingAction#CANCEL}, then the re-pricing is skipped.   If the {@link RepriceProcessingAction} of the re-price form is {@link RepriceProcessingAction#DEFER}, then re-pricing is again deferred to a later stage. @param deferredRepriceAction the Deferred Re-price Action to be processed @param domainCategoryCode the Domain Category Code @param repriceDate the Re-price Date @return {@link ProcessingStatusType#COMPLETED} if the re-pricing takes place, {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				ProcessingStatusType#NOT_APPLICABLE} if there-pricing is cancelled, {@link ProcessingStatusType#NOT_STARTED} if the re-pricing is deferred again.
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to re price/refreshed. In what manner it is to be re priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to re price/refreshed. In what manner it is to be re priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	@see com.ofss.fc.domain.pricing.entity.repricing.AbstractRepriceEvent#processRepriceEvent(java.lang.String, com.ofss.fc.domain.pricing.context.PricingEventContext, com.ofss.fc.domain.pricing.entity.repricing.RepriceDetails, com.ofss.fc.datatype.Period)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	This method calls respective module to get stage details. @param arrangementNumber @param priceType @param domainCategoryId @param stageDetails @param dueDate @param elementCode @return
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be re-priced/refreshed. In what manner it is to be re-priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	Validates Event Price Variances @param domainId @param domainCategoryId @param accountId @param eventPriceVariances



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRepricingConfiguration	The concrete method implementation will be provided by each servicing module. There they can write their own rule and decide if the account is to be reprice/refreshed. In what manner it is to be re priced. Pricing will read the return form dto and re-price the account as instructed by the servicing module. @param arrangementNumber @param emptyForm @param repriceDate @return {@link RepriceFormDTO} @throws FatalException	This method performs the task of filling the repriceFormDTO so as to indicate to Pricing as to in what manner the account is to rePriced/refreshed. Depending on the return form dto, Pricing will rePrice/refresh the account accordingly. This method first checks whether the RateChange is eligible or not by calling the isRateChangeEligible() method. If it is eligible, it logs the corresponding RateChange action in an actions table, from which, the action would be picked up during the Batch for processing. If it is not eligible, then the RejectedEntry details are logged in a <RejectedEntry> table for internal audit purposes. Based on the eligibility of the Ratechange action & the rules from the <RepriceRulesTable>, the repriceFormDTO will be filled & returned. @param accountId @param repriceDate @param emptyForm @return RepriceFormDTO
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyDomainFactChange		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logLoanRateChangeAction	This method is used to log rate change action in loans module, whenever rate change happens in pricing, due to UDM linkage @param accountId Loan account id @param priceType {@link PriceType} @param repriceDate Date from which reprice is effective @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} containing details of UDMs linked to account @param pricingInitiated @throws FatalException	This method will be called after account or deposit is opened for adding rate variances to the pricing plan after account/deposit is opened i.e with arrangement number/account number @param sessionContext : holds the basic information about the user of the system @param arrangementNumber : the arrangement number of the party @param domain : the domain name like Banking, Insurance @param domainCategory : the domain category name like CASA, TD @param eventRateVarianceDTOs : array of EventRateVarianceDTO which the encapsulates the rate variance information for a particular event @return TransactionStatus : holds information on the transaction status @throws FatalException if the pricing contract service throws a FatalException @author: SureshRagala
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logLoanRateChangeAction	This method is used to log rate change action in loans module, whenever rate change happens in pricing, due to UDM linkage @param accountId Loan account id @param priceType {@link PriceType} @param repriceDate Date from which reprice is effective @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} containing details of UDMs linked to account @param pricingInitiated @throws FatalException	This method will be called after account or deposit is opened for adding rate variances to the pricing plan after account/deposit is opened i.e with arrangement number/account number. This should be used <b>while UDM addition is not done by ELPBC screen</b> . @param sessionContext : holds the basic information about the user of the system @param arrangementNumber : the arrangement number of the party @param domain : the domain name like Banking, Insurance @param domainCategory : the domain category name like CASA, TD @param eventRateVarianceDTOs : array of EventRateVarianceDTO which the encapsulates the rate variance information for a particular event @return TransactionStatus : holds information on the transaction status @throws FatalException if the pricing contract service throws a FatalException @author: SudipG

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	logLoanRateChangeAction	This method is used to log rate change action in loans module, whenever rate change happens in pricing, due to UDM linkage @param accountId Loan account id @param priceType {@link PriceType} @param repriceDate Date from which reprice is effective @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} containing details of UDMs linked to account @param pricingInitiated @throws FatalException	@see com.ofss.fc.app.adapter.pricing.IPricingAccountAdapter#logLoanRateChangeAction(java.lang.String, com.ofss.fc.enumeration.pricing.PriceType, com.ofss.fc.datatype.Date, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO[])
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	fetchBundleExpiryDateForAccount	This method will give bundle expiry date for an account. It will be null if the bundle is not limited period or bundle is not subscribed for an account. @param accountId unique identifier of an account @return BundleExpiryDate	This method will give bundle expiry date for an account. It will be null if the bundle is not limited period or bundle is not subscribed for an account. @param accountId unique identifier of an account @return BundleExpiryDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	fetchBundleExpiryDateForAccount	This method will give bundle expiry date for an account. It will be null if the bundle is not limited period or bundle is not subscribed for an account. @param accountId unique identifier of an account @return BundleExpiryDate	Helper function that calls the {@link com.ofss.fc.domain.pricing.service.price.PriceComputationService PriceComputationService} for each of the {@link PriceOfferingElement PriceOfferingElements} within the set argument. It adds all the ProcessedBasePrices and applies the campaign variance to get the net amount, in the form of a MoneyBag containing all the currencies specified in the currencyCodes, which is then set to the totalPrice of the {@link PriceFeature}. @param pricingContext the pricing context @param arrangementNumber the arrangement number @param eventCode the event code @param effectiveDate the effective date @param priceOfferings the set of {@link OfferingElement OfferingElements} @param currencyCodes the string array of currency codes @param useUnauthorizedVariancesFlag the unauthorised variance flag @return the {@link PriceFeature} @throws FatalException
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	fetchCounterDetails	This Adapter is used to get details of counter for specific arrangement for event codes for given period of begin and end dates.  Modules which want to avail the functionality of Counter based concessions like Fee Group Waiver has to provide implementation of this interface. @param counterDetailsRequestDTO {@link CounterDetailsRequestDTO} containing input details required to get counter information @return {@link CounterDetailsDTO} containing details of counter related information	This methods returns the cheapest/costliest/earliest #count {@link FlattenedTransactionDetails} from the total FlattenedTransactionDetails.  It will create a {@link Comparator<FlattenedTransactionDetails>} to sort the given list and return the first #count from this list @param arrangementNumber the Arrangement number @param feeConcessionPlan the {@link FeeConcessionGroupPlan} @param feeConcessionGroup the {@link FeeConcessionGroup} @param accountCurrency Account currency @param transactions the list of {@link FlattenedTransactionDetails} @return the cheapest #count {@link FlattenedTransactionDetails}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	fetchCounterDetails	This Adapter is used to get details of counter for specific arrangement for event codes for given period of begin and end dates.  Modules which want to avail the functionality of Counter based concessions like Fee Group Waiver has to provide implementation of this interface. @param counterDetailsRequestDTO {@link CounterDetailsRequestDTO} containing input details required to get counter information @return {@link CounterDetailsDTO} containing details of counter related information	This method is used to match counter in pricing and in account module  If mismatch is detected alert will be sent to concern person. @param arrangementNumber @param cycleStartDate @param cycleEndDate @param transactions
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyRepriceResult	This method notifies result to caller after reprice in {@link RepriceResultDTO}. @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate	
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyRepriceResult	This method notifies result to caller after reprice in {@link RepriceResultDTO}. @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate	This method notifies result to caller after reprice in {@link RepriceResultDTO}. @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyRepriceResult	This method notifies result to caller after reprice in <a href="#">{@link RepriceResultDTO}</a> . @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate	This method processes the given deferred re-price action. It calls the respective adapter (depending on the given domain category code) to fetch the re-price configuration information <a href="#">{@link RepriceFormDTO}</a> .   If the <a href="#">{@link RepriceProcessingAction}</a> of the re-price form is <a href="#">{@link RepriceProcessingAction#ACCEPT}</a> , then the pricing plan is re-priced for the applicable pricing element.   If the <a href="#">{@link RepriceProcessingAction}</a> of the re-price form is <a href="#">{@link RepriceProcessingAction#REJECT}</a> or <a href="#">{@link RepriceProcessingAction#CANCEL}</a> , then the re-pricing is skipped.   If the <a href="#">{@link RepriceProcessingAction}</a> of the re-price form is <a href="#">{@link RepriceProcessingAction#DEFER}</a> , then re-pricing is again deferred to a later stage. @param deferredRepriceAction the Deferred Re-price Action to be processed @param domainCategoryCode the Domain Category Code @param repriceDate the Re-price Date @return <a href="#">{@link ProcessingStatusType#COMPLETED}</a> if the re-pricing takes place, <a href="#">{@link ProcessingStatusType#NOT_APPLICABLE}</a> if there-pricing is cancelled, <a href="#">{@link ProcessingStatusType#NOT_STARTED}</a> if the re-pricing is deferred again.
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyRepriceResult	This method notifies result to caller after reprice in <a href="#">{@link RepriceResultDTO}</a> . @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate	This method is used to notify the LCM on the new computation date as a result of the reprice.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	notifyRepriceResult	This method notifies result to caller after reprice in {@link RepriceResultDTO}. @param arrangementNumber @param repriceFormDTO @param repriceResultDTO @param repriceDate	Validates Event Price Variances @param domainId @param domainCategoryId @param accountId @param eventPriceVariances
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or ifThis method is called before any other method in this service @author: SureshRag
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan .  This should be used <b><em>while UDM addition is not done by ELPBC screen</em></b>. @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or ifThis method is called before any other method in this service @author: SudipG

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	It's calling from ContractApplicationService: mergePriceVariance(SessionContext, PricingContractContextDTO, EventRateVarianceDTO[]) to persist the recommendations done for rateVariance in pricing recommendation @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event.
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	This method will validate and throws business exception if following conditions are not fulfilled. <pre> <strong>Display error if the UDM start date is:</strong> <li>Before First Disbursement Date.</li> <li>Before Partial Write Off date </li> <li>Before Full Write Off date. </li> <li>Before Loan Recession. </li> <li>Before Conversion date (i.e. Movement from Legacy to New System) </li> <li>Back value date should not be a holiday. </li> <li>Account should not be in fixed rate on back value date. </li> </pre> @param accountId :: String @param eventVarianceDTOArray :: EventVarianceDTO[]
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	This method will validate and throws business exception if following conditions are not fulfilled. <pre> <strong>Display error if the UDM start date is:</strong> <li>Before First Disbursement Date.</li> <li>Before Partial Write Off date </li> <li>Before Full Write Off date. </li> <li>Before Loan Recession. </li> <li>Before Conversion date (i.e. Movement from Legacy to New System) </li> <li>Back value date should not be a holiday. </li> <li>Account should not be in fixed rate on back value



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				date. </li> </pre> @param accountId @param varainceStartDate
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	validateRateVariance	Includes the module specific validations for Rate Variances @param arrangementNumber @param eventVarianceDTOArray @throws FatalException	This method validates the variances @param domain @param domainCategory @param arrangementNumber @param eventVarianceDTOs @throws FatalException
IPricingAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pricing\IPricingAccountAdapter	getRateTypeForAccount	This method will get the {@link RateType} for an account. @param accountId @param domainCategory @param evaluationDate @return {@link RateType} for an account @throws FatalException	To get the rate type for the given arrangement number and domain category @param arrangementNumber @param domainCategory @param evaluationDate @return rateType
ILoanLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMAAdapter	checkFacilityAvailability	<p>This method checks whether the Facility for provided Facility Id is available or not.  If the facility is not available, it raises an error message for the same. </p> @param facilityId @param borrowingEntityId @throws FatalException	<p>This method checks whether the Facility for provided Facility Id is available or not.  If the facility is not available, it raises an error message for the same. </p> @param facilityId @param borrowingEntityId @throws FatalException
ILoanLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMAAdapter	checkFacilityAvailability	<p>This method checks whether the Facility for provided Facility Id is available or not.  If the facility is not available, it raises an error message for the same. </p> @param facilityId @param borrowingEntityId @throws FatalException	This method checks if the mockery is enabled or not and based on that it performs the further processing. @param String @return ILoanLimitsApplicationAdapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkFacilityAvailability	<p>&lt;p&gt;This method checks whether the Facility for provided Facility Id is available or not.&lt;br&gt; If the facility is not available, it raises an error message for the same. &lt;/p&gt;</p> <p>@param facilityId @param borrowingEntityId @throws FatalException</p>	<p>&lt;p&gt;This method is used to perform validations with respect to facility Details. &lt;br&gt; It performs following validations during Loan Account Opening:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To make sure whether facility is available or not.&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Account creation date resides in between Facility Period Dates&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount&lt;/li&gt;</li> </ul> <p>&lt;br&gt; If above tests are successful, then it performs below mentioned tasks:-</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To link Loan Account to Facility.&lt;/li&gt;</li> <li>&lt;li&gt;To raise an Earmark for the Loan Account.&lt;/li&gt;</li> </ul> <p>@param applicationContext @param loanAccountId @throws FatalException</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateAccountUtilization	<p>This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility.</p> <p>@param sessionContext @param utilizationDTO</p>	<p>This method updates the loan amount with the proposed amount at the account level @param sessionContext @param scheduleId @param loanAccountProposedAmount @param loanApprovedAmountHistoryApplicationService @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. For Loans , the update service will be used to increase/decrease the credit facility linked to it. @param utilizationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. For Loans , the update service will be used to increase/decrease the credit facility linked to it. @param utilizationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	This method checks if the mockery is enabled or not and based on that it performs the further processing. @param String @return ILoanLimitsApplicationAdapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	<pre>This method will update the principal utilization, if any for the loan account. </pre> @param utilizationRequestDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	The method will increase amount in TotalPremiuAmount and premiumAmount in LoanAccountBalances, reschedule the account, pass accounting entries, mark LCMUtilization. @param applicationContext @param accountId @param valueDate @param premiumAmount @param premiumFrequencyType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.INTERNAL.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMAadapter	updateAccountUtilization	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facilities are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	<pre>This method is used for posting the disbursement transaction. Posting the disbursement involves a) Logging Loan Ledger entries for disbursement b) Invoking Settlement Module for credit of the disbursement amount to the beneficiary account (GL, CASA, EXTERNAL) c) Incrementing the Loan balances d) Increasing the disbursement count on the account e) Posting Deductions f) Logging various batch actions (Accelerated Payment, Account Payment Instruction, Advance Alert) e) Invoking interest module for registering the interest plan &#064;param applicationContext &#064;param eventId &#064;param accountId &#064;param disbursementAmount &#064;param feeCollectionDTO &#064;param componentPreferenceDTO @param scheduleIdForPrincipalAmount TODO @param scheduleIdForPeriodicFees TODO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	@param applicationContext @param eventId @param accountId @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	This method is used to perform backdated subsequent disbursement. @param applicationContext @param loanAccount @param eventId @param deductionModeType @param wrappedPriceFeatureDTO @param disbursementData @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMAadapter	updateAccountUtilization	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	<pre>This method performs the doSubsequentDisbursement operation. It performs the following steps: <ol><li>doRescheduleOnSubsequentDisbursement</li><li>validateSchedule</li><li>generateRepaymentRecordsOnReschedule</li><li>saveScheduleOnReschedule</li><li>post the Disbursement Transaction</li><li>raise Accounting</li></ol> #param applicationContext #param accountId #param savedScheduleArrangementId #param disbursementDate #param topupDisbursementAmount #param capitalizeArrears #param feeCollectionDTO #param eventData #param amountTagToDirectionMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateAcco untUtilizati on	This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans, the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param utilizationDTO	This method will be called to credit loan account. Following steps will be carried out in this method. 1)RPA will be credited with the given amount   2)Log Interest Computation Action.  3)Fetch Reschedule OPTion Type from loan product parameters  4)Appropriate RPA Balance.  5)Update Account Utilization. @param sessionContext @param accountId @param valueDate @param eventId @param debitAmount @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacil ityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	This method is used to update the loan account title @param accountId @param accountTitle

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacil ityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	This method is used to amend the account title Internally calls com.ofss.fc.domain.loan.entity.account.LoanAccount.updateLoanAccountTitle(String) which updates the title @param sessionContext @param loanAmendAccountDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacil ityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacil ityLinkage	This method is used to update the details of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. Validations are also as part of the update service to ensure that the facility are valid and only accounts linked to the facility can utilize. For Loans , the update service will be used to increase/decrease the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. @param sessionContext @param facilityLinkageDTO	<pre>This method will update the principal utilization, if any for the loan account. </pre> @param utilizationRequestDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	reverseAcc ountUtilizat ion	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. The repaid amount will also be handled accordingly. @param sessionContext @param utilizationDTO	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. @param utilizationDTO @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	reverseAcc ountUtilizat ion	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. The repaid amount will also be handled accordingly. @param sessionContext @param	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. @param utilizationDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			utilizationDTO	
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	reverseAcc ountUtilizat ion	This method is used to reverse a utilization of the account which was performed earlier. Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. The repaid amount will also be handled accordingly. @param sessionContext @param utilizationDTO	This method checks if the mockery is enabled or not and based on that it performs the further processing. @param String @return ILoanLimitsApplicationAdapter

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	reverseAcc ountUtilizat ion	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. The repaid amount will also be handled accordingly. @param sessionContext @param utilizationDTO	<pre>This method will reverse the principal utilization, if any for the loan account. </pre> @param utilizationRequestDTO
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	reverseAcc ountUtilizat ion	This method is used to reverse a utilization of the account which was performed earlier.Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. For CASA, based on the amounts and the fund utilization sequence of the account, different facilities can be utilized like Credit Facility (OD), Uncleared Funds Facility, Temporary Excess Facility. The repaid amount will also be handled accordingly. @param sessionContext @param	<pre>This method is responsibe for reversal of disbursement Reversal of disbursement will involve deleting all data inserted during the disbursement process &#064;param applicationContext &#064;param accountId &#064;param originalTransactionReferenceNumber @param componentPreferenceId @param isSettlementPayoutReversal @return void

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			utilizationDTO	
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	createFacilit yEARMark	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException	This method is used for amending(increasing/decreasing) the loan amount. It accepts the loan current and proposed sanctioned amount, validates if increase/decrease in amount is allowed considering the current rate type of the loan account and update {@link LoanApprovedAmountHistory} and sanctioned amount maintained at account level with the proposed term amount. It does the following, 1. Validate the System Constraints //Create LoanAmendAccountSystemConstraintsDTO with eventType=AmendEventType.AMOUNT_CHANGE, LoanAmendAccountSystemConstraint and validate the constraints via LoanAmendAccountSystemConstraint.isSatisfiedBy() 2. Determine the rate type of the loan i.e Fixed or Variable //Load LoanAccountAdditionalDetails via LoanAccountAdditionalDetailsRepository.getInstance().load(accountId) method //Compare currentDate with LoanAccountAdditionalDetails.fixedPeriodEndDate if(currentDate<fixedPeriodEndDate) loanRateType=LoanRateType.FIXED else loanRateType=LoanRateType.VARIABLE 2. Fetch the customer loan account via LoanAccountService.fetchLoanAccount(accountId) method. 3. Fetch the loan product preferences associated with the customer loan account via LoanProductService.fetchLoanProductPreferences() method. 4. Create BusinessPolicyDTO based on increase/decrease in loan amount and validate the policy //Increase in loan amount if(loanAccountProposedAmount >

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre>loanAccountCurrentAmount) { //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.INCREASE_AMOUNT } else { //Decrease in loan amount //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.DECREASE_AMOUNT } //Create LoanAmendAccountBusinessPolicy and Validate the policy via LoanAmendAccountBusinessPolicy.validate() method. 5. Update the LoanApprovedAmountHistory with the proposed term amount via LoanApprovedAmountHistoryApplicationService.up dateLoanApprovedAmountHistory() method; This method internally updates the sanctionedAmount maintained at account level. @param sessionContext @param scheduleId @param loanAccountCurrentAmount @param loanAccountProposedAmount</pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	createFacilit yEARMark	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	createFacilit yEARMark	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException	This method is used to delete an Earmark.   It is being used during Loan Account Closure. @param facilityEarMarkingDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	createFacilit yEARMark	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException	<p>This method is used to perform validations with respect to facility Details.   It performs following validations during Loan Account Opening:-   <ul> <li>To make sure whether facility is available or not.</li> <li>To make sure whether Loan Account creation date resides in between Facility Period Dates</li> <li>To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount</li> </ul>   If above tests are successful, then it performs below mentioned tasks:-   <ul> <li>To link Loan Account to Facility.</li> <li>To raise an Earmark for the Loan Account.</li> </ul> @param applicationContext @param loanAccountId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacilit yEARMar k	This method is used to update an Earmark.   It is being used during Loan Account Amend. @param facilityEarMarkingDTO @return @throws FatalException	This method is used for amending(increasing/decreasing) the loan amount. It accepts the loan current and proposed sanctioned amount, validates if increase/decrease in amount is allowed considering the current rate type of the loan account and update {@link LoanApprovedAmountHistory} and sanctioned amount maintained at account level with the proposed term amount. It does the following, 1. Validate the System Constraints //Create LoanAmendAccountSystemConstraintsDTO with eventType=AmendEventType.AMOUNT_CHANGE, LoanAmendAccountSystemConstraint and validate the constraints via LoanAmendAccountSystemConstraint.isSatisfiedBy() 2. Determine the rate type of the loan i.e Fixed or Variable //Load LoanAccountAdditionalDetails via LoanAccountAdditionalDetailsRepository.getInstance( ).load(accountId) method //Compare currentDate with LoanAccountAdditionalDetails.fixedPeriodEndDate



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> if(currentDate&lt;fixedPeriodEndDate) loanRateType=LoanRateType.FIXED else loanRateType=LoanRateType.VARIABLE 2. Fetch the customer loan account via LoanAccountService.fetchLoanAccount(accountId) method. 3. Fetch the loan product preferences associated with the customer loan account via LoanProductService.fetchLoanProductPreferences() method. 4. Create BusinessPolicyDTO based on increase/decrease in loan amount and validate the policy //Increase in loan amount if(loanAccountProposedAmount &gt; loanAccountCurrentAmount) { //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.INCREASE_AMOUNT } else //Decrease in loan amount //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.DECREASE_AMOUNT } //Create LoanAmendAccountBusinessPolicy and Validate the policy via LoanAmendAccountBusinessPolicy.validate() method. 5. Update the LoanApprovedAmountHistory with the proposed term amount via LoanApprovedAmountHistoryApplicationService.up dateLoanApprovedAmountHistory() method; This method internally updates the sanctionedAmount maintained at account level. @param sessionContext @param scheduleId @param loanAccountCurrentAmount @param loanAccountProposedAmount </pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	updateFacil ityEARMar k	This method is used to update an Earmark.   It is being used during Loan Account Amend. @param facilityEarMarkingDTO @return @throws FatalException	This method is used to update an Earmark.   It is being used during Loan Account Amend. @param facilityEarMarkingDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	deleteFacilit yEARMark	This method is used to delete an Earmark.   It is being used during Loan Account Closure. @param facilityEarMarkingDTO @throws FatalException	This method is used to delete an Earmark.   It is being used during Loan Account Closure. @param facilityEarMarkingDTO @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	closeFacilit yEarmark	This method is used to close an earmark for a facility.   @param facilityEarmarkReferenceNumber @throws FatalException	<p>This method will close the loan account on the inputed date.  Following actions are being performed by this method:-  <ol> <li>Update Loan Account status as "Closed"</li> <li>Update Loan Account Closure Date</li> <li>Update reason for Loan Account Closure</li> <li>Update Central Bank Report details</li> <li>Update Bundle Subscription of Loan Account</li> <li>Update Asset Classification</li> <li>Delink Loan Account from Facility</li> <li>Remove Offset Linkages</li> </ol> </p> @param applicationContext @param account @param processDate @param eventId TODO
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	linkFacility ToLoanAcc ount	This method is used to link the Facility to Loan Account.   @param accountId @param facilityId @throws FatalException	This method is used to link the Facility to Loan Account.   @param accountId @param facilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	linkFacility ToLoanAcc ount	This method is used to link the Facility to Loan Account.   @param accountId @param facilityId @throws FatalException	<p>This method is used to perform validations with respect to facility Details.   It performs following validations during Loan Account Opening:-   <ul> <li>To make sure whether facility is available or not.</li> <li>To make sure whether Loan Account creation date resides in between Facility Period Dates</li> <li>To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount</li> </ul>   If above tests are successful, then it performs below mentioned tasks:-   <ul> <li>To link Loan Account to Facility.</li> <li>To raise an Earmark for the Loan Account.</li> </ul> @param applicationContext @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				loanAccountId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	deLinkLoan AccountFrom Facility	This method is used to de-link Loan Account from facility.   @param accountId @param facilityId @throws FatalException	<p>This method is responsible for delinking Loan Account from facility.  During Loan Account Closure via Early & Final Settlement,This method is invoked to delink the Loan Account from facility to which it was linked (while opening the account).</p> @param loanAccount @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	deLinkLoan AccountFrom Facility	This method is used to de-link Loan Account from facility.   @param accountId @param facilityId @throws FatalException	This method is used to de-link Loan Account from facility.   @param accountId @param facilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	deLinkLoan AccountFrom Facility	This method is used to de-link Loan Account from facility.   @param accountId @param facilityId @throws FatalException	<p>This method is responsible for delinking Loan Account from facility.  During Loan Account Closure via Early & Final Settlement,This method is invoked to delink the Loan Account from facility to which it was linked (while opening the account).</p> @param loanAccount @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	checkDisbu rsementDat eAgainstFa cilityAvail mentDates	This method is used during Loan Disbursement process.   It checks the following aspects:-   <ol> <li>At least first disbursement should have been done before Facility - Availment Expiry Date</li> <li>No disbursement is allowed after Facility - Last Availment Expiry Date</li> </ol> @param instructionNumber @param disbursementExecutionDate @param facilityId @throws FatalException	This method is used during Loan Disbursement process.   It checks the following aspects:-   <ol> <li>At least first disbursement should have been done before Facility - Availment Expiry Date</li> <li>No disbursement is allowed after Facility - Last Availment Expiry Date</li> </ol> @param instructionNumber @param disbursementExecutionDate @param facilityId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkDisbursementDateAgainstFacilityAvailmentDates	This method is used during Loan Disbursement process.   It checks the following aspects:-   <ol> <li>At least first disbursement should have been done before Facility - Availment Expiry Date</li> <li>No disbursement is allowed after Facility - Last Availment Expiry Date</li> </ol> @param instructionNumber @param disbursementExecutionDate @param facilityId @throws FatalException	This method validates whether auto disbursement can be done on the account by applying the business policies @param context @param disbursementData @return boolean @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkDisbursementDateAgainstFacilityAvailmentDates	This method is used during Loan Disbursement process.   It checks the following aspects:-   <ol> <li>At least first disbursement should have been done before Facility - Availment Expiry Date</li> <li>No disbursement is allowed after Facility - Last Availment Expiry Date</li> </ol> @param instructionNumber @param disbursementExecutionDate @param facilityId @throws FatalException	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	validateAgainstFacilityAmounts	<p>This method validates the following:-   <ul> <li>Checks whether Loan Approved Amount is less than Facility sanctioned amount or not.</li> </ul> </p> @param loanApprovedAmount @param facilityId @throws FatalException	<p>This method validates the following:-   <ul> <li>Checks whether Loan Approved Amount is less than Facility sanctioned amount or not.</li> </ul> </p> @param loanApprovedAmount @param facilityId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	validateAga instFacility Amounts	<p>This method validates the following:-   <ul> <li>Checks whether Loan Approved Amount is less than Facility sanctioned amount or not.</li> </ul> </p> @param loanApprovedAmount @param facilityId @throws FatalException	@param applicationContext @param disbursementAmount @param loanAccount @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	validateAga instFacility Amounts	<p>This method validates the following:-   <ul> <li>Checks whether Loan Approved Amount is less than Facility sanctioned amount or not.</li> </ul> </p> @param loanApprovedAmount @param facilityId @throws FatalException	<pre>This method is used for posting the disbursement transaction. Posting the disbursement involves a)Logging Loan Ledger entries for disbursement b)Incrementing the Loan balances c)Increasing the disbursement count on the account d)Posting Deductions e)Logging various batch actions (Accelerated Payment , Account Payment Instruction , Advance Alert) f)Invoking interest module for registering the interest plan &#064;param applicationContext &#064;param eventId &#064;param accountId &#064;param disbursementAmount &#064;param feeCollectionDTO &#064;param accountingEventAmountTagDetailsDTOs @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	validateAga instFacility Amounts	<p>This method validates the following:-   <ul> <li>Checks whether Loan Approved Amount is less than Facility sanctioned amount or not.</li> </ul> </p> @param loanApprovedAmount @param facilityId @throws FatalException	<p>This method is used to perform validations with respect to facility Details.   It performs following validations during Loan Account Opening:-   <ul> <li>To make sure whether facility is available or not.</li> <li>To make sure whether Loan Account creation date resides in between Facility Period Dates</li> <li>To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount</li> </ul>   If above tests are successful, then it performs below mentioned tasks:-   <ul> <li>To link Loan Account to Facility.</li> <li>To raise an Earmark for the Loan Account.</li> </ul> @param applicationContext @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				loanAccountId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	validateAga instFacility PeriodDate s	<p>&lt;p&gt;This method ensures whether Current Process Date is in between Facility Period Dates. Following things are being ensured:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Whether Facility can be utilized or not.&lt;/li&gt;</li> <li>&lt;li&gt;Whether facility has been expired or not.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; Following conditions are being checked:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Check if Process Date &gt; Facility Start Date&lt;/li&gt;</li> <li>&lt;li&gt;Check if Process Date &lt; Facility End Date&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @param processDate @param facilityId @throws FatalException</p>	<p>&lt;p&gt;This method ensures whether Current Process Date is in between Facility Period Dates. Following things are being ensured:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Whether Facility can be utilized or not.&lt;/li&gt;</li> <li>&lt;li&gt;Whether facility has been expired or not.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; Following conditions are being checked:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Check if Process Date &gt; Facility Start Date&lt;/li&gt;</li> <li>&lt;li&gt;Check if Process Date &lt; Facility End Date&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @param processDate @param facilityId @throws FatalException</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	validateAga instFacility PeriodDate s	<p>&lt;p&gt;This method ensures whether Current Process Date is in between Facility Period Dates. Following things are being ensured:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Whether Facility can be utilized or not.&lt;/li&gt;</li> <li>&lt;li&gt;Whether facility has been expired or not.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; Following conditions are being checked:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Check if Process Date &gt; Facility Start Date&lt;/li&gt;</li> <li>&lt;li&gt;Check if Process Date &lt; Facility End Date&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @param processDate @param facilityId @throws FatalException</p>	<p>&lt;p&gt;This method is used to perform validations with respect to facility Details. &lt;br&gt; It performs following validations during Loan Account Opening:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To make sure whether facility is available or not.&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Account creation date resides in between Facility Period Dates&lt;/li&gt;</li> <li>&lt;li&gt;To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;br&gt; If above tests are successful, then it performs below mentioned tasks:- &lt;br&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;To link Loan Account to Facility.&lt;/li&gt;</li> <li>&lt;li&gt;To raise an Earmark for the Loan Account.&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; @param applicationContext @param loanAccountId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	validateMaturityDateAgainstFacilityDates	<p>This method validates the following:-   <ul> <li>Check whether Loan maturity Date is not exceeding Facility Term Date.   Facility End Date has been considered for this.</li> </ul> @param loanMaturityDate @param facilityId @throws FatalException	<p>This method validates the following:-   <ul> <li>Check whether Loan maturity Date is not exceeding Facility Term Date.   Facility End Date has been considered for this.</li> </ul> @param loanMaturityDate @param facilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkCovenantStatus	<p>This method validates the following:-   <ul> <li>Check whether status of Covenant (which is linked to the Facility) is "ACTIVE" or not.</li> </ul> </p> @param borrowingEntityId @param facilityId @throws FatalException	<p>This method validates the following:-   <ul> <li>Check whether status of Covenant (which is linked to the Facility) is "ACTIVE" or not.</li> </ul> </p> @param borrowingEntityId @param facilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	@param applicationContext @param loanAmendAccountDTO
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityDetails	This method is use to fetch a facility. It does by invoking {@link FacilityApplicationService#fetchFacility()} method. @param facilityId @return facilityDTO	<pre>This method is called to fetch fetchMainLine's productCode. The following steps are carried out in this method:- &lt;br&gt; 1)MainLine Id is fetched by calling fetchMainFacilityForAccount method of LimitsAccountApplicationService(). 2)MainLine details are fetched by calling method LoanLCMA dapter.fetchFacilityDetails(String). 3)Then mainline's productGroupCode is fetched from the response. 4)Call is given to fetchProductGroupDetails(ApplicationContext, String) to check whether the mainline's productgroup is MOF or NOT. 5)If mainline's product group is MOF then product group code of mainline is returned. </pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	This method is used to raise an Earmark.   It is being used during Loan Account Opening. @param facilityEarMarkingDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	@param applicationContext @param changeOverDraftToTermLoanRequestDTO @return ChangeOverDraftToTermLoanResponse
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	addFacility	This method is use to add a facility. It does by invoking {@link FacilityApplicationService#addFacility()} method. @param facilityDTO @return transactionStatus	<pre>This method is used to perform the ChangeTermLoanToOverDraftTrnsaction. &lt;br&gt;The following steps are carried out in this method:-&lt;br&gt; 1) System Constraints are checked.&lt;br&gt; 2)Business Policy's are checked. 3) Total Due amount of the LoanAccount is fetched by calling method fetchTotalDueAmount()&lt;br&gt; 4) EFS of the Loan Amount is performed by calling method performLoanClosure by passing TotalDueAmount .&lt;br&gt; 5) New Facility is created .This is done by calling addFacility method of lcm module via LoanLCM adapter.&lt;br&gt; 6) OverDraft linkage is attached to transaction account by calling method attachODLinkage. </pre> @param sessionContext @param LendingVariationRequestDTO @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is used to update the loan account title @param accountId @param accountTitle

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\com\ofss\fc\app\adapter \loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is used for amending(increasing/decreasing) the loan amount. It accepts the loan current and proposed sanctioned amount, validates if increase/decrease in amount is allowed considering the current rate type of the loan account and update {@link LoanApprovedAmountHistory} and sanctioned amount maintained at account level with the proposed term amount. It does the following, 1. Validate the System Constraints //Create LoanAmendAccountSystemConstraintsDTO with eventType=AmendEventType.AMOUNT_CHANGE, LoanAmendAccountSystemConstraint and validate the constraints via LoanAmendAccountSystemConstraint.isSatisfiedBy() 2. Determine the rate type of the loan i.e Fixed or Variable //Load LoanAccountAdditionalDetails via LoanAccountAdditionalDetailsRepository.getInstance().load(accountId) method //Compare currentDate with LoanAccountAdditionalDetails.fixedPeriodEndDate if(currentDate<fixedPeriodEndDate) loanRateType=LoanRateType.FIXED else loanRateType=LoanRateType.VARIABLE 2. Fetch the customer loan account via LoanAccountService.fetchLoanAccount(accountId) method. 3. Fetch the loan product preferences associated with the customer loan account via LoanProductService.fetchLoanProductPreferences() method. 4. Create BusinessPolicyDTO based on increase/decrease in loan amount and validate the policy //Increase in loan amount if(loanAccountProposedAmount > loanAccountCurrentAmount) { //Create LoanAmendAccountBusinessPolicyDTO with

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>eventType=AmendEventType.INCREASE_AMOUNT } else { //Decrease in loan amount //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.DECREASE_AMOUNT } //Create LoanAmendAccountBusinessPolicy and Validate the policy via LoanAmendAccountBusinessPolicy.validate() method.</p> <p>5. Update the LoanApprovedAmountHistory with the proposed term amount via LoanApprovedAmountHistoryApplicationService.updateLoanApprovedAmountHistory() method; This method internally updates the sanctionedAmount maintained at account level. @param sessionContext @param scheduleId @param loanAccountCurrentAmount @param loanAccountProposedAmount</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is used to amend the account title Internally calls com.ofss.fc.domain.loan.entity.account.LoanAccount.updateLoanAccountTitle(String) which updates the title @param sessionContext @param loanAmendAccountDTO
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is used to update an Earmark.   It is being used during Loan Account Amend. @param facilityEarMarkingDTO @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateFacility	This method is use to update a facility. It does by invoking {@link FacilityApplicationService#updateFacility()} method. @param facilityDTO @return transactionStatus	<pre>This method will update the principal utilization, if any for the loan account. </pre> @param utilizationRequestDTO
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	This method is responsible for fetching the entire details of the   Loan Account . The inputs to the method is accountId .   All the details associated with the loan account such as dates (Account Opening Date ,   Maturity Date ) , Status , Customer Information etc are returned back to the callee.  @param accountId @return LoanAccountCommonInformationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchFacilit yInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	This method is used for amending(increasing/decreasing) the loan amount. It accepts the loan current and proposed sanctioned amount, validates if increase/decrease in amount is allowed considering the current rate type of the loan account and update {@link LoanApprovedAmountHistory} and sanctioned amount maintained at account level with the proposed term amount. It does the following, 1. Validate the System Constraints //Create LoanAmendAccountSystemConstraintsDTO with eventType=AmendEventType.AMOUNT_CHANGE, LoanAmendAccountSystemConstraint and validate the constraints via LoanAmendAccountSystemConstraint.isSatisfiedBy() 2. Determine the rate type of the loan i.e Fixed or Variable //Load LoanAccountAdditionalDetails via LoanAccountAdditionalDetailsRepository.getInstance( ).load(accountId) method //Compare currentDate with LoanAccountAdditionalDetails.fixedPeriodEndDate if(currentDate<fixedPeriodEndDate) loanRateType=LoanRateType.FIXED else loanRateType=LoanRateType.VARIABLE 2. Fetch the customer loan account via LoanAccountService.fetchLoanAccount(accountId) method. 3. Fetch the loan product preferences associated with the customer loan account via LoanProductService.fetchLoanProductPreferences() method. 4. Create BusinessPolicyDTO based on increase/decrease in loan amount and validate the policy //Increase in loan amount if(loanAccountProposedAmount > loanAccountCurrentAmount) { //Create LoanAmendAccountBusinessPolicyDTO with

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> eventType=AmendEventType.INCREASE_AMOUNT } else { //Decrease in loan amount //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.DECREASE_AMOUNT } //Create LoanAmendAccountBusinessPolicy and Validate the policy via LoanAmendAccountBusinessPolicy.validate() method. 5. Update the LoanApprovedAmountHistory with the proposed term amount via LoanApprovedAmountHistoryApplicationService.up dateLoanApprovedAmountHistory() method; This method internally updates the sanctionedAmount maintained at account level. @param sessionContext @param scheduleId @param loanAccountCurrentAmount @param loanAccountProposedAmount </pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchFacilit yInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	<p>&lt;p&gt;This method validates the following:- &lt;br&gt; &lt;ul&gt; &lt;li&gt;Check whether Loan maturity Date is not exceeding Facility Term Date. &lt;br&gt; Facility End Date has been considered for this.&lt;/li&gt; &lt;/ul&gt; @param loanMaturityDate @param facilityId @throws FatalException</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchFacilit yInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	This method is used during Loan disbursement instruction set-up process.   It fetches availment expiry date and last availment expiry date from LCM.  @param facilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchFacilit yInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	<p>This method is used to perform validations with respect to facility Details.   It performs following validations during Loan Account Opening:-   <ul> <li>To make sure whether facility is available or not.</li> <li>To make sure whether Loan Account creation date resides in between Facility Period Dates</li> <li>To make sure whether Loan Approved Amount is less than Facility Sanctioned Amount</li> </ul>   If above tests are successful, then it performs below mentioned tasks:-   <ul> <li>To link Loan Account to Facility.</li> <li>To raise an Earmark for the Loan Account.</li> </ul> @param applicationContext @param loanAccountId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityInfo	This method is used to fetch facility information. @param creditFacilityId @return FacilityInquiryResponse	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchMainFacilityForAccount	This method is used to mainline facility details of the given loan Account . @param creditFacilityId @return FacilityInquiryResponse	This method is used to mainline facility details of the given loan Account . @param creditFacilityId @return FacilityInquiryResponse
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchMainFacilityForAccount	This method is used to mainline facility details of the given loan Account . @param creditFacilityId @return FacilityInquiryResponse	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchMainF acilityForA ccount	This method is used to mainline facility details of the given loan Account . @param creditFacilityId @return FacilityInquiryResponse	<pre>This method is called to fetch fetchMainLine's productCode. The following steps are carried out in this method:- &lt;br&gt; 1)MainLine Id is fetched by calling fetchMainFacilityForAccount method of LimitsAccountApplicationService(). 2)MainLine details are fetched by calling method LoanLCMAadapter.fetchFacilityDetails(String). 3)Then mainline's productGroupCode is fetched from the response. 4)Call is given to fetchProductGroupDetails(ApplicationContext, String) to check whether the mainline's productgroup is MOF or NOT. 5)If mainline's product group is MOF then product group code of mainline is returned. </pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchFacilit yAvailment Dates	This method is used during Loan disbursement instruction set-up process.   It fetches availment expiry date and last availment expiry date from LCM.  @param facilityId @throws FatalException	This method is used during Loan disbursement instruction set-up process.   It fetches availment expiry date and last availment expiry date from LCM.  @param facilityId @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchFacilityEarMarkingDetails	This method is use to fetch a facility ear marking details. It does by invoking {@link FacilityEarMarkingApplicationService#fetchFacilityEarMarking()} method. @param facilityEarMarkingDTO @return FacilityEarMarkingInquiryResponse	This method is used for amending(increasing/decreasing) the loan amount. It accepts the loan current and proposed sanctioned amount, validates if increase/decrease in amount is allowed considering the current rate type of the loan account and update {@link LoanApprovedAmountHistory} and sanctioned amount manintained at account level with the proposed term amount. It does the following, 1. Validate the System Constraints //Create LoanAmendAccountSystemConstraintsDTO with eventType=AmendEventType.AMOUNT_CHANGE, LoanAmendAccountSystemConstraint and validate the constraints via LoanAmendAccountSystemConstraint.isSatisfiedBy() 2. Determine the rate type of the loan i.e Fixed or Variable //Load LoanAccountAdditionalDetails via LoanAccountAdditionalDetailsRepository.getInstance().load(accountId) method //Compare currentDate with LoanAccountAdditionalDetails.fixedPeriodEndDate if(currentDate<fixedPeriodEndDate) loanRateType=LoanRateType.FIXED else loanRateType=LoanRateType.VARIABLE 2. Fetch the customer loan account via LoanAccountService.fetchLoanAccount(accountId) method. 3. Fetch the loan product preferences associated with the customer loan account via LoanProductService.fetchLoanProductPreferences() method. 4. Create BusinessPolicyDTO based on increase/decrease in loan amount and validate the policy //Increase in loan amount if(loanAccountProposedAmount > loanAccountCurrentAmount) { //Create LoanAmendAccountBusinessPolicyDTO with

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> eventType=AmendEventType.INCREASE_AMOUNT } else { //Decrease in loan amount //Create LoanAmendAccountBusinessPolicyDTO with eventType=AmendEventType.DECREASE_AMOUNT } //Create LoanAmendAccountBusinessPolicy and Validate the policy via LoanAmendAccountBusinessPolicy.validate() method. 5. Update the LoanApprovedAmountHistory with the proposed term amount via LoanApprovedAmountHistoryApplicationService.up dateLoanApprovedAmountHistory() method; This method internally updates the sanctionedAmount maintained at account level. @param sessionContext @param scheduleId @param loanAccountCurrentAmount @param loanAccountProposedAmount </pre>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	isFacilityIs OverLined	This method will return true if facility of the given account is overlined and false if facility is not overlined. @param sessionContext @param accountId @return isFacilityIsOverLined: boolean @throws FatalException	This method will return true if facility of the given account is overlined and false if facility is not overlined. @param sessionContext @param accountId @return isFacilityIsOverLined: boolean @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchCollate ralsForBorr owingEntit y	This service will fetch the collaterals linked with the borrowing entity Id. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String borrowingEntityId @return {@link FacilityInquiryResponse} @throws FatalException	<pre>This method will fetch the collaterals for the borrowing entity, if any for the loan account. </pre> @param borrowingEntityId @return {@link FacilityInquiryResponse}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchCollateralsForFacilityId	This method will fetch the collaterals for the facility Id, if any for the loan account. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,String facilityId @return {@link FacilityInquiryResponse} @throws FatalException	<pre>This method will fetch the collaterals for the facility Id, if any for the loan account. </pre> @param facilityId @return {@link FacilityInquiryResponse}
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchAccountsForFacility	This method is used to obtain the accounts that are linked to the Facility. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param facilityId @return {@link AccountLinkageResponse} @throws FatalException	This method is used to obtain the accounts that are linked to the Facility. @param sessionContext @param creditFacilityId @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	updateBalance		This method is used to update the utilization of the account linked to the facility. Based on the input, system will decide whether the facility utilization is increased/decreased. For Loans , the update service will be used to increase/decrease the credit facility linked to it. @param utilizationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	updateBala nce		<pre> This method will interact with the LoanELCMA dapter, which interacts with ELCM module to update the utilization made by the loan account. &lt;ol&gt; &lt;li&gt;load LoanBalanceService and get balance details after posting.&lt;/li&gt; &lt;li&gt;make sessionContext from applicationContext.&lt;/li&gt; &lt;li&gt;prepare the UtilizationRequestDTO by calling prepareUtilizationRequest().&lt;/li&gt; &lt;li&gt;send the UtilizationRequestDTO to ELCM through LoanELCMA dapter for updating the utilization by the loan account.&lt;/li&gt; &lt;li&gt;throw an exception, in case there is an error.&lt;/li&gt; &lt;/ol&gt; </pre> <p>@param applicationContext @param accountId @param facilityId @param prePrincipalBalance @param preInterestBalance @param preFeesBalance</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	updateBala nce		<pre> This method will update the principal utilization, if any for the loan account. </pre> <p>@param utilizationRequestDTO</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	reverseBala nce		<p>This method is used to reverse a utilization of the account which was performed earlier. Based on the input, system will decide whether the facility utilization has to be increased/decreased. For Loans , the reversal will update only the increase/decrease of the credit facility linked to it. @param utilizationDTO @throws FatalException</p>
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	reverseBala nce		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	reverseBalance		
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	reverseBalance		<pre>This method will reverse the principal utilization, if any for the loan account. </pre> @param utilizationRequestDTO
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	isAccountLinkedToFacilityFees	The method is responsible to inquire if given account is nominated for facility level fees. @param accountId @return @throws FatalException	The method is responsible to inquire if given account is nominated for facility level fees. @param accountId @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	isAccountLinkedToFacilityFees	The method is responsible to inquire if given account is nominated for facility level fees. @param accountId @return @throws FatalException	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkIfLastAccountUnderFacility	@param accountId @param facilityId @return @throws FatalException	This method will find out if the given account can be closed.  The method will return true if   1. Account is not linked to any active policy.  2. The periodicity of the linked active policy is One-Time. @param accountId @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkIfLastAccountUnderFacility	@param accountId @param facilityId @return @throws FatalException	The method is responsible to return amount to be refunded by insurance provider in case account is settled earlier than the maturity date. The method will  1. Check if given account is linked to any active policy.  2. If given account is linked to active policy, what is its premium paying period.  3. If periodicity of the policy is One-Time, inquire refund amount from insurance provider. @param accountId @param currencyCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	checkIfFacilityIsAvai led	This method is for checking the facility availment and expiry dates while opening an account under a facility and its subsequent disbursements. The logic of the method is now present in LCM instead of Loans. @param facilityId @param valueDate @return	@param applicationContext @param disbursementAmount @param loanAccount @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	checkIfFacilityIsAvai led	This method is for checking the facility availment and expiry dates while opening an account under a facility and its subsequent disbursements. The logic of the method is now present in LCM instead of Loans. @param facilityId @param valueDate @return	This method validates all the business policies for disbursement instructions set-up  It will validate the instruction set-up date, execution date, instruction amount etc.  @param loanDisbursementInstructionDTO @param loanAccount @param totalDisbursementAmount @param maintenanceType @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	checkIfFacilityIsAvai led	This method is for checking the facility availment and expiry dates while opening an account under a facility and its subsequent disbursements. The logic of the method is now present in LCM instead of Loans. @param facilityId @param valueDate @return	
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMA dapter	fetchBorrowingEntity DetailsForId	<pre>This method will fetch the borrowing entity details for the borrowing entity Id passed. </pre> @param borrowingEntityId @return {@link BorrowingEntityInquiryResponse}	This method is responsible for fetching the basic details of the   Loan Account . The inputs to the method is accountId .   @param accountId @return LoanAccountCommonInformationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	fetchBorrowingEntityDetailsForId	<pre>This method will fetch the borrowing entity details for the borrowing entity Id passed. </pre> @param borrowingEntityId @return {@link BorrowingEntityInquiryResponse}	<pre>This method will fetch the borrowing entity details for the borrowing entity Id passed. </pre> @param borrowingEntityId @return {@link BorrowingEntityInquiryResponse}
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkIfAccountLinkedToFacility	The method will check if the account provided is nominated for facility fees as on facility date. @param accountId @param valueDate @return @throws FatalException	The method will return facility fees to be levied on account closure. @param accountId @param valueDate @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	checkIfAccountLinkedToFacility	The method will check if the account provided is nominated for facility fees as on facility date. @param accountId @param valueDate @return @throws FatalException	The method will return if given account is/was nominated for facility fees. @param accountId @param valueDate @return @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	confirmAccountClosure	The method will inform LCM on account closure. @param accountId @param valueDate @throws FatalException	
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	confirmAccountClosure	The method will inform LCM on account closure. @param accountId @param valueDate @throws FatalException	The method will inform LCM on account closure. @param applicationContext @param accountId @param valueDate @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanLCMA dapter	chargeFacilityFeesOnHardshipreliefExpiry	The method is responsible to levy facility fees which were deffered during hardship relief period. @param accountId @throws FatalException	The method is responsible to levy facility fees which were deffered during hardship relief period. @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	fetchOutsta ndingCredi tExposure	@param sessionContext @param facilityId @return @throws FatalException	
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	checkIfCon ditionsAre Complied	@param sessionContext @param facilityId @return @throws FatalException	@param applicationContext @param disbursementAmount @param loanAccount @throws FatalException
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	checkIfCon ditionsAre Complied	@param sessionContext @param facilityId @return @throws FatalException	This method will check the conditions are compiled or not via LCM. 
ILoanLCMA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanLCMAadapt er	checkIfCon ditionsAre CompliedD uringInstru ction	@param sessionContext @param facilityId @return @throws FatalException	This method will check the conditions are compiled or not via LCM. 
ILinkedAcco untBalanceFr omTDAdapt er	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \collection\ILinkedAcco untBalanceFromTDAdapt er	fetchAllDep osits		
ILinkedAcco untBalanceFr omTDAdapt er	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \collection\ILinkedAcco untBalanceFromTDAdapt er	fetchAllDep osits		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyAdapter	fetchPartyDetails		Appservice method to fetch Party Details from OBP database.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to load the borrowing entity id for the given account @param accountId @return String
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	The parties present in the linkages are also verified.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to test the facility account linkage during the creation of the facility. @param facilityType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	The parties present in the linkages are also verified.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to pre populate the borrowing entity and the facility id's in the map. This is done because BE and Facility relationship needs to be established. @param borrowingEntityList @param notationDtlsList
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This creates a map of fact id and fact values. @param creditFacility @return Map<String, Object>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This creates a map of fact id and fact values. @param creditFacilityDTO @return Map<String, Object>
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This represents the fetchFacilitySummary method of the entity. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,Filter[],a ,a TODO: Explain the filter above @return {@link FacilityInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getBorrowingEntityId	This method is used to load the borrowing entity id for the given account @param accountId @return String	This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getDemandDepositAccount	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getDemandDepositAccount	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String	Method to fetch the demand deposit account. @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getDemandDepositAccount	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getDemandDepositAccount	This method is used to load the Demand Deposit Account for fetching the borrowing entity id and currency code for a particular account id. @param accountId @return String	This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	getLimitAmountForUnclearedFundFacility	This method is used to get Limit amount For Uncleared Fund Facility for the given account id. @param sessionContext @param accountCode @return limitAmount @throws FatalException	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAccountDetails	This method return the module code to which the account belongs to @param accountId @return @throws FatalException	This method is used to fetch the details of the account to be used in the exposure tracking summary. @param accountIdArray @return @since 2.3
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAccountDetails	This method return the module code to which the account belongs to @param accountId @return @throws FatalException	This method is used to populate the account level details required for the credit exposure document generation. All the facilities upto the account level are shown in this along with account level details. @param referenceId @param requestDtls @param finalFacilitySet @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAllAccountsForBorrowingEntity	This method calls the party module to fetch all the accounts belonging to a borrowing entity.  The borrowing entity may contain a single party or more than one parties. This method utilizes the party and account relationship established in party module to fetch all the accounts for all the parties belonging to a borrowing entity. @param partyIds @param com.ofss.fc.enumeration.TaskModuleType @return @throws FatalException	This method calls the party module to fetch all the accounts belonging to a borrowing entity.  The borrowing entity may contain a single party or more than one parties. This method utilizes the party and account relationship established in party module to fetch all the accounts for all the parties belonging to a borrowing entity. @param partyIds @param com.ofss.fc.enumeration.TaskModuleType @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAllAccountsForBorrowingEntity	This method calls the party module to fetch all the accounts belonging to a borrowing entity.  The borrowing entity may contain a single party or more than one parties. This method utilizes the party and account relationship established in party module to fetch all the accounts for all the parties belonging to a borrowing entity. @param partyIds @param com.ofss.fc.enumeration.TaskModuleType @return @throws FatalException	This fetches all the accounts related to given borrowing entity.  The service resolves the accounts for the borrowing entity in the following way: <ul> <li>First the party/parties which form the borrowing entity are fetched.</li> <li>The accounts related to those party/parties are fetched using the party and account relationship.</li> <li>After retrieving the account id's the basic account details are fetched from the respective account module.</li> </ul> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,borrowingEntity, , This parameter captures the borrowing entity id.  The borrowing entity is used to derive the party/parties linked to the facility. @fcb.param MI,TaskModuleType,taskModuleType, , This parameter captures the task module type.  The module to which the account belongs to  @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				com.ofss.fc.app.adapter.account.dto.AccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAllAccountsForBorrowingEntity	This method calls the party module to fetch all the accounts belonging to a borrowing entity.  The borrowing entity may contain a single party or more than one parties. This method utilizes the party and account relationship established in party module to fetch all the accounts for all the parties belonging to a borrowing entity. @param partyIds @param com.ofss.fc.enumeration.TaskModuleType @return @throws FatalException	This fetches all the accounts related to given party/parties.  The service resolves the accounts for the party/parties in the following way: <ul><li>The accounts related to those party/parties are fetched using the party and account relationship.</li><li>After retrieving the account id's the basic account details are fetched from the respective account module.</li></ul> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String[],partyIds, , This parameter captures an array of party id.  These are the parties linked to the facility. @fcb.param MI,TaskModuleType,taskModuleType, , This parameter captures the task module type.  The module to which the account belongs to  @return {@link com.ofss.fc.app.adapter.account.dto.AccountInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAllAccountsForBorrowingEntity	This method calls the party module to fetch all the accounts belonging to a borrowing entity.  The borrowing entity may contain a single party or more than one parties. This method utilizes the party and account relationship established in party module to fetch all the accounts for all the parties belonging to a borrowing entity. @param partyIds @param com.ofss.fc.enumeration.TaskModuleType @return @throws FatalException	This fetches all the loan accounts present in the facility heirarchy</br> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,facilityId, , This parameter captures the facility id.  @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccountForFacilityCreation	This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to validate the account during the temp excess and uncleared creation/update. @param accountId @param facilityType
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	@param appContext @param creditFacility @param computationDate @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to check whether the facility has any deferred fee. In case of deferred fees, the facility cannot be closed.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchDDAccount	Fetches the details of the Demand Deposit Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method return the module code to which the account belongs to @param accountId @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	@param appContext @param creditFacility @param computationDate @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This method is used to check whether the facility has any deferred fee. In case of deferred fees, the facility cannot be closed.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchLoanAccount	Fetches the details of the Loan Account. @param accountId @return com.ofss.fc.app.adapter.account.dto.AccountDTO	This fetches all the loan accounts present in the facility heirarchy</br> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,facilityId, , This parameter captures the facility id.  @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	processFees	@param applicationContext @param feesChargingRequest @return @throws FatalException	@param applicationContext @param feesChargingRequest @param eventId @param chargingDate @param periodicPriceFeatureDTOArray @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	processFees	@param applicationContext @param feesChargingRequest @return @throws FatalException	This method is used to fetch the price and charge the nominated account. Also the plan will be closed once the fees are charged. @param facilityId @param evaluationDate @param closeOrExpiryDate @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	processFees	@param applicationContext @param feesChargingRequest @return @throws FatalException	This method is used to process the deferred fees for the facility if any. Applicable only for facilities in which loan account is attached. @param facilityId @param currencyCode @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	processFees	@param applicationContext @param feesChargingRequest @return @throws FatalException	@param applicationContext @param feesChargingRequest @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	isAccountInDeadStatus	This method is used to check whether the demand deposit account is in dead status or not. @param nominatedAccount @return	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	isAccountInDeadStatus	This method is used to check whether the demand deposit account is in dead status or not. @param nominatedAccount @return	This method is used to filter the accounts that are valid from the given list. The list can contain both Loan and CASA accounts. @param acctIdList, The list which contains all the accounts ids. Currently the list might contain loans as well as CASA. @return List<String>, The output list of valid accounts. @since 2.3
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	fetchAccountDetailsForExposure	This method is used to fetch the details of the account to be used in the exposure tracking summary. @param accountIdArray @return Map<String, ExposureTrackingInquiryResponse>	This method is used to populate the account level details required for the credit exposure document generation. All the facilities upto the account level are shown in this along with account level details. @param referenceId @param requestDtls @param finalFacilitySet @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateLoanAcctForFees	This method checks whether a given loan account is valid for fees debit. @param accountId @return boolean	@param accountDTO @return
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	filterValidAccountsFromList	This method is used to filter the accounts that are valid from the given list. The list can contain both Loan and CASA accounts. @param acctIdList @return @since 2.3	This method is used to check whether this is the last account that is linked in the facility tree. This needs to be a tree because the fees are created at the Ultimate Parent facility and this might be one of the sub facility. Hence the entire tree needs to be checked if it is the last account of the tree. If yes, return true indicating the fees for it will be collected. In case there are multiple accounts present , do not collect any fees for the closure. @param facilityId @param accountId @since 2.3 @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateDDAcctForFees	This method checks whether a given DD account is valid for fees debit. @param accountId @return boolean @since 2.3	@param accountDTO @return
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	isAccountInFeeHardshipRelief	This method returns if account is under fee hardship relief or not. i.e. Can Facility fee be charged from the account or not. @ {@link Ref com.ofss.fc.app.adapter.impl.bundle.BundleDemandDepositAdapter.isAccountInHardship(String)} @param accountId @since 2.3 @return	This method checks whether a given DD account is valid for fees debit. @param accountId @return boolean @since 2.3

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	isAccountInFeeHardshipRelief	This method returns if account is under fee hardship relief or not. i.e. Can Facility fee be charged from the account or not. @ {@link Ref com.ofss.fc.app.adapter.impl.bundle.BundleDemandDepositAdapter.isAccountInHardship(String)} @param accountId @since 2.3 @return	This method returns if account is under fee hardship relief or not. i.e. Can Facility fee be charged from the account or not. @ {@link Ref com.ofss.fc.app.adapter.impl.bundle.BundleDemandDepositAdapter.isAccountInHardship(String)} @param accountId @since 2.3 @return
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	isAccountInFeeHardshipRelief	This method returns if account is under fee hardship relief or not. i.e. Can Facility fee be charged from the account or not. @ {@link Ref com.ofss.fc.app.adapter.impl.bundle.BundleDemandDepositAdapter.isAccountInHardship(String)} @param accountId @since 2.3 @return	This method is used to check whether the CASA account is valid currently. If not fee processing will not happen for such accounts. Applicable only for Demand Deposit accounts. @param nominatedAccount @param accountModuleType @return @throws FatalException
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation(). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to test the facility account linkage during the creation of the facility. @param facilityType
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation(). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return	This method is used to validate the account during the temp excess and uncleared creation/update. @param accountId @param facilityType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation( ). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to check if the account for the future dated facility is eligible for creation.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation( ). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to check if the account for the future dated facility is eligible for creation.
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation( ). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation( ). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to validate the account during the temp excess and uncleared creation/update. @param accountId @param facilityType
ILimitsAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsAccountAdapter	validateAccount	This method is used to fetch the account details and ensure that is valid for temporary excess creation. This is replica of the service validateAccountForFacilityCreation( ). This service excludes borrowing entity validation. @param sessionContext @param accountId @param facilityType @return @throws FatalException	This method is used to fetch the account details and ensure that is valid for temporary excess creation. @param sessionContext @param accountId @param facilityType @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchMaxAmountLimit		This method first interacts with account inquiry service to fetch the product code and then fetches the max amount limit per channel by interacting with the product management service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to fetch the product code @param accountType the account type of the transaction @param currencyCode the currency code to be used @return Money the max limit of the transaction per channel

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchMaxAmountLimit		This method will check for the maximum limit of the transaction amount for the channel under the product @param channelContext @param partyID @param accountID @param transactionValue
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchMaxAmountLimit		This method will return the adapter class from the factory class, this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	This operation enables you to perform an earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @param MI, POSChannelContext, channelContext, , POSChannelContext which must contain the POS Key.  @param MI, ISOBusinessContext, data, , BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount, its currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status. The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRE_01}</li> </ul> Transaction requires



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>authorisation for one of the reasons mentioned below</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I_B_WDR_LIM}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt;</p> <p>&lt;b&gt;Transaction failure&lt;/b&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOIDD}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOIDD}&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYNC_ERROR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FATAL}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt;</li> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS}</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>UFF_FUNDS}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#RECOR D_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSB K_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B LOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODR_ ONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_M ANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {<a href="#">@link</a> com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#SUCCESS&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL&lt;/li&gt; &lt;li&gt;<a href="#">@link</a> com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	Prepare response for OFFUS transaction by populating ATMBalanceResponse object. @fcb.param MI,TransactionStatus,status, , The status of the transaction which indicates success or failure @return BaseResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p> </li></ul>
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	This method raises the account event for given channel @param sessionContext the session context to be used in the request @param channelAccountingEvent the accounting event DTO to be used in raising accounting event @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	This method applies Acquirer and Forwarder Fee charges to a CASA account for a transaction of Instrument.   After Debiting the Customer Account by Total Fee amount, Acquirer and Forwarder GL's are credited by Acquirer and Forwarder amount respectively. Acquirer and Forwarder Fee may or may not be present for the given transaction.  @fcb.param MI,SessionContext, sessionContext, , This parameter captures the basic information about the user of the system. @fcb.param MI,FinancialContext,txnValue, , Financial Context holds basic information about the transaction.  @fcb.param MI,String,accountId,1..16, The CASA account number, for which Cash deposit transaction has to be performed @fcb.param MI,double,transactionAmount, , The amount in terms of the transaction currency @fcb.param MI,int,mnemonic, , Mnemonic code of the transaction @fcb.param MI,int,processingCode, , The processing code under which this transaction will be   registered in the system @fcb.param MI,int,transactionType, , Can be one of the values from <ul> <li>{@link AbstractApplication#NORMAL_TRANSACTION}</li> <li>{@link AbstractApplication#ADVICE_TRANSACTION}</li> </ul> @fcb.param MI,TransactionStatus,status, , The status of the transaction which indicates success or failure @return BaseResponse @throws FatalException @throws OverridableException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	Prepare response for OFFUS transaction by populating ATMBalanceResponse object. @fcb.param MI,TransactionStatus,status, , The status of the transaction which indicates success or failure @return BaseResponse
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseAccountingEvent	This method raises the account event for given channel @param channelAccountingEvent @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchProduct	This method fetches the product code using the account number and the account type @param accountID @param accountType @return @throws FatalException	This method first interacts with account inquiry service to fetch the product code and then fetches the max amount limit per channel by interacting with the product management service @param sessionContext the session context to be used in interacting with the service @param accountID the account id to fetch the product code @param accountType the account type of the transaction @param currencyCode the currency code to be used @return Money the max limit of the transaction per channel
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchProduct	This method fetches the product code using the account number and the account type @param accountID @param accountType @return @throws FatalException	This method fetches the product code using the account number and the account type @param sessionContext the session context to be used in the request @param accountID the account id to get overdraft preference against @param accountType the account type to be used in request @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchProduct	This method fetches the product code using the account number and the account type @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchPOSTerminalLedgerId	This method fetches the POS terminal ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchATMCashLedgerId	This method fetches the ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method fetches the ATM cash ledger code for a given fact in PMU @param sessionContext the session context to be used in the request @param glName the glname to be used @param terminalId the terminal against which GL is to be fetched @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchATMCashLedgerId	This method fetches the ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchForwarderCashLedgerId	This method fetches the Forwarder cash ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method fetches the Forwarder cash ledger code for a given fact in PMU @param sessionContext the session context to be used in the request @param glName the glname to be used @param forwarderId the forwarder against which GL is to be fetched @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchForwarderCashLedgerId	This method fetches the Forwarder cash ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchForwarderFeeLedgerId	This method fetches the Forwarder fee ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method fetches the Forwarder fee ledger code for a given fact in PMU @param sessionContext the session context to be used in the request @param glName the glname to be used @param forwarderId the forwarder against which GL is to be fetched @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchForwarderFeeLedgerId	This method fetches the Forwarder fee ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by {@link AccountingEventApplicationService #raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus}.  <ol> AccountingBehaviourDTO has two attributes <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> <ol> @param transactionReferenceNumber @param originalTransactionReferenceNumber	This operation enables you to perform a earmark removal against a POS Terminal GL  by passing the account to be debited and corresponding GL to be credited  as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.  @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate.  @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons    <b>Success</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</li> </ul>   <b>Transaction failure due to warning</b> <ul> <li>{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRESENT_01}  Transaction requires authorisation for one of the reasons mentioned below <ul> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</li> <li>{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_ACCT_CODE}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#OLTP_INT_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_INFO_CHANGED}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link</li> <li>com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt;</li> </ul> <p>@throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">{@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus}</a>.  <ol style="list-style-type: none"> <li>AccountingBehaviourDTO has two attributes <a href="#">&lt;li&gt;isAuthorized: by default value as TRUE;</a></li> <li><a href="#">&lt;li&gt;isBalancingRequired: by default value is FALSE</a></li> </ol> </p> <p><a href="#">@param transactionReferenceNumber</a>  <a href="#">@param originalTransactionReferenceNumber</a></p>	<p>This operation enables you to perform an earmark removal against a POS Terminal GL by passing the account to be debited and corresponding GL to be credited as input. <a href="#">@fcb.param MI,POSChannelContext,channelContext, , POS Channel Context</a> which must contain the POS Key.  <a href="#">@fcb.param MI,ISOBusinessContext,data, ,BusinessContext</a> Context holds basic information about the transaction. It contains the monetary value of the amount, its currency and the currency conversion rate.  <a href="#">@return</a> response An instance of <a href="#">{@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse}</a> containing the required information along with status. The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons</p> <p><b>Success</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</a></li> </ul> <p><b>Transaction failure due to warning</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}</a> Transaction requires authorisation for one of the reasons mentioned below</li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</a></li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">{@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus}</a>.  <ol style="list-style-type: none"> <li>AccountingBehaviourDTO has two attributes <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </li></ol> </p> <p><b>@param transactionReferenceNumber</b>  <b>@param originalTransactionReferenceNumber</b></p>	<p>This operation enables you to perform a earmark removal against a POS Terminal GL by passing the account to be debited and corresponding GL to be credited as input. <b>@fcb.param MI,POSChannelContext,channelContext, , POS Channel Context</b> which must contain the POS Key.  <b>@fcb.param MI,ISOBusinessContext,data, ,BusinessContext</b> Context holds basic information about the transaction. It contains the monetary value of the amount , it's currency and the currency conversion rate.  <b>@return</b> response An instance of <a href="#">{@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse}</a> containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons  <ul style="list-style-type: none"> <li><b>Success</b></li> <li><a href="#">{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}</a></li> <li>Transaction failure due to warning</li> <li><a href="#">{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}</a> Transaction requires authorisation for one of the reasons mentioned below <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}</a></li> <li><a href="#">{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</a></li> </ul> </li> </ul> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TransactionStatus</a>.  AccountingBehaviourDTO has two attributes <a href="#">isAuthorized</a>: by default value as TRUE;  <a href="#">isBalancingRequired</a>: by default value is FALSE  @<a href="#">param transactionReferenceNumber</a>  @<a href="#">param originalTransactionReferenceNumber</a></p>	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TransactionStatus</a>.  AccountingBehaviourDTO has two attributes <a href="#">isAuthorized</a>: by default value as TRUE;  <a href="#">isBalancingRequired</a>: by default value is FALSE  @<a href="#">param transactionReferenceNumber</a>  @<a href="#">param originalTransactionReferenceNumber</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	<p>The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">{@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TrasactionStatus}</a>.  <ol style="list-style-type: none"> <li>AccountingBehaviourDTO has two attributes <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </li></ol> </p> <p><b>@param</b> transactionReferenceNumber  <b>@param</b> originalTransactionReferenceNumber</p>	<p>This method reverses Acquirer and Forwarder Fee charges to a CASA account for a transaction of Instrument. After Debiting the Customer Account by Total Fee amount, Acquirer and Forwarder GL's are credited by Acquirer and Forwarder amount respectively. Acquirer and Forwarder Fee may or may not be present for the given transaction.</p> <p><b>@fcb.param</b> MI,SessionContext, sessionContext, , This parameter captures the basic information about the user of the system.  <b>@fcb.param</b> MI,FinancialContext,txnValue, , Financial Context holds basic information about the transaction.  <b>@fcb.param</b> MI,String,accountId,1..16, The CASA account number, for which Cash deposit transaction has to be performed  <b>@fcb.param</b> MI,double,transactionAmount, , The amount in terms of the transaction currency  <b>@fcb.param</b> MI,int,mnemonic, , Mnemonic code of the transaction  <b>@fcb.param</b> MI,int,processingCode, , The processing code under which this transaction will be registered in the system  <b>@fcb.param</b> MI,int,transactionType, , Can be one of the values from <ul style="list-style-type: none"> <li><a href="#">{@link AbstractApplication#NORMAL_TRANSACTION}</a></li> <li><a href="#">{@link AbstractApplication#ADVICE_TRANSACTION}</a></li> </ul> <b>@fcb.param</b> MI,TransactionStatus,status, , The status of the transaction which indicates success or failure  <b>@return</b> BaseResponse  <b>@throws</b> FatalException  <b>@throws</b> OverridableException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	raiseReverseAccountingEvent	The method will reverse the accounting event raised for the original transaction reference number. The AccountingBehaviourDTO required by <a href="#">@link AccountingEventApplicationService#raiseReverseAccountingEventForTransaction(transactionReferenceNumber:String, originalTransactionReferenceNumber: String, inputDTO:AccountingBehaviourDTO)::TransactionStatus</a> . <ol style="list-style-type: none"> <li>isAuthorized: by default value as TRUE;</li> <li>isBalancingRequired: by default value is FALSE</li> </ol> @params transactionReferenceNumber originalTransactionReferenceNumber	This method will return the adapter class from the factory class, this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelProductAdapter	getSavingsPreferenceDto	This method returns the savings preferences DTO for a given account id and account type @params sessionContext @param accountID @param accountType @return @throws FatalException	This operation allows for cheque book request for number of cheque leaves to be issued, no of cheque book to be issued These details are returned as a data transfer object of type ATMBalanceResponse @params fcb.param MI, ATMChannelContext, channelContext, , ATMChannelContext which must contain the ATMKey. @params fcb.param MI, ISOBusinessContext, data, , BusinessContext holds basic information about the transaction. It contains the monetary value of the amount, its currency and the currency conversion rate. @return ATMBalanceResponse A response

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>object which holds the transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li>Success</li> <li>Transaction failure</li> <li>Transaction failure</li> <li>Error Reason may be one of the following <ul style="list-style-type: none"> <li>ACCT_NT_FLG_CLS</li> <li>ACCT_INFO_CHANGED</li> <li>NO_OUTSTANDING_PO</li> <li>ACCT_CLOSED_TODAY</li> <li>ACCT_BAL_NT_ZERO</li> <li>MID_ECH_INVALID_ACCT_NO</li> <li>MID_EUT_INVALID_USERID</li> <li>MID_EST_INV_BANK</li> </ul> </li> </ul> <p><i>@throws FatalException Please refer to the documentation above for Transaction failure section</i></p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	getSavingsPreferenceDto	This method returns the savings preferences DTO for a given account id and account type @param sessionContext @param accountID @param accountType @return @throws FatalException	This method returns the savings preferences DTO for a given account id and account type @param sessionContext the session context to be used in the request @param accountID the account id to get savings preference against @param accountType the account type to be used in request @param productCode the product code to which account number belongs to @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	getSavingsPreferenceDto	This method returns the savings preferences DTO for a given account id and account type @param sessionContext @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class, this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchLedgerCodeForDomainCategory	This method fetches the Acquirer fee ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method fetches the Acquirer fee ledger code for a given fact in PMU @param sessionContext the session context to be used in the request @param acctRole the account role to be used @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchLedgerCodeForDomainCategory	This method fetches the Acquirer fee ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method applies Acquirer and Forwarder Fee charges to a CASA account for a transaction of Instrument.   After Debiting the Customer Account by Total Fee amount, Acquirer and Forwarder GL's are credited by Acquirer and Forwarder amount respectively. Acquirer and Forwarder Fee may or may not be present for the given transaction.  @fcb.param MI,SessionContext, sessionContext, , This parameter captures the basic information about the user of the system. @fcb.param MI,FinancialContext,txnValue, , Financial Context holds basic information about the transaction.  @fcb.param MI,String,accountId,1..16, The CASA account number, for which Cash deposit transaction has to be performed @fcb.param MI,double,transactionAmount, , The amount in terms of the transaction currency @fcb.param MI,int,mnemonic, , Mnemonic code of the transaction @fcb.param MI,int,processingCode, , The processing code under which this transaction will be   registered in the system @fcb.param MI,int,transactionType, , Can be one of the values from <ul> <li>{@link AbstractApplication#NORMAL_TRANSACTION}</li> <li>{@link AbstractApplication#ADVICE_TRANSACTION}</li> </ul> @fcb.param MI,TransactionStatus,status, , The status of the transaction which indicates success or failure @return BaseResponse @throws FatalException @throws OverridableException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchLedgerCodeForDomainCategory	This method fetches the Acquirer fee ledger code for a given fact in PMU @param accountID @param accountType @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchDomainCategoryFromEvent	This method returns true if the the category is domain for given event id ,false otherwise @param sessionContext @param eventId @return @throws FatalException	This method raises the account event for given channel @param sessionContext the session context to be used in the request @param channelAccountingEvent the accounting event DTO to be used in raising accounting event @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchDomainCategoryFromEvent	This method returns true if the the category is domain for given event id ,false otherwise @param sessionContext @param eventId @return @throws FatalException	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH,AT etc. @param eventId @return @throws FatalException
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchDomainCategoryFromEvent	This method returns true if the the category is domain for given event id ,false otherwise @param sessionContext @param eventId @return @throws FatalException	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of fees.
IChannelProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelProductAdapter	fetchDomainCategoryFromEvent	This method returns true if the the category is domain for given event id ,false otherwise @param sessionContext @param eventId @return @throws FatalException	This method will return the adapter class from the factory class,this method also checks if the mock is enabled and returns the mock data if mock is enabled
IDSLBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLBropAdapter	fetchDenominationsForGivenCurrency	This method will fetch denominations for a given currency by interfacing with Common Service. @param currencyCode @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLBropAdapter	fetchDenominationsForGivenCurrency	This method will fetch denominations for a given currency by interfacing with Common Service. @param currencyCode @return @throws FatalException	The following method is used to validate the transaction amount and denomination. @param currencyDenominationDTO @param moneyDTO @throws FatalException
IDSLBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLBropAdapter	fetchNotificationForCustomerDashboard	<p>This method will fetch notification when request is made from Customer Dashboard. </p> @param sessionContext @param accountNumber @param partyID @return @throws FatalException	
IAssetClassificationRecoveryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationRecoveryAdapter	doOnlineSettlement	This method performs online settlement for the given amount.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @param shortNarration The short description for transaction @param offerId The source event for transaction @return {@link AssetClassificationSettlementResponse}	This method performs online settlement for the given amount.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @param shortNarration The short description for transaction @param offerId The source event for transaction @return {@link AssetClassificationSettlementResponse} Object containing details of settlement

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			nseDTO} Object containing details of settlement	
IAssetClassificationRecoveryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationRecoveryAdapter	doOnlineSettlement	This method performs online settlement for the given amount.  @param eventCode The event code which will be raised for accounting @param productCode The product id @param componentPreferenceDTO {@link ComponentPreferenceDTO} Object which contains information required to perform settlement like settlement direction, settlement mode etc @param amount The amount for which settlement needs to be done @param settlementValueDate Value date for settlement execution @param narration The description for online transactions @param messageType {@link MessageType} used for settlement @param shortNarration The short description for transaction @param offerId The source event for transaction @return {@link AssetClassificationSettlementResponseDTO} Object containing details of settlement	This method triggers accounting event, updates Written Off recovered amount and inserts Written Off recovery

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	publishRule	This service will publish the rule for given rule definition. @param sessionContext,ruleDefinition @throws FatalException	This represents the addRelationshipRule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,RelationshipRuleDTO,relationshipRuleDTO, ,relationshipRuleDTO TODO: Explain the relationshipRuleDTO above @return (@link com.ofss.fc.service.response.TransactionStatus) @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	publishRule	This service will publish the rule for given rule definition. @param sessionContext,ruleDefinition @throws FatalException	This service will publish the rule for given rule definition. @param sessionContext,ruleDefinition @throws FatalException
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	viewRule	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId, effectiveDate @throws FatalException	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId, effectiveDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	viewLatestRuleForProcessDate	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId @throws FatalException	This represents the fetchRelationshipRule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,RelationshipRuleDTO,relationshipRuleDTO, ,relationshipRuleDTO TODO: Explain the relationshipRuleDTO above @return (@link com.ofss.fc.app.party.dto.relation.rule.RelationshipRuleInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	viewLatestRuleForProcessDate	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId @throws FatalException	
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	viewLatestRuleForProcessDate	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId @throws FatalException	This service will inquire for the existing rule for given rule id. @param sessionContext, ruleId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	modifyRule	This service will modify the existing rule for given rule id. @param sessionContext, ruleDefinition @throws FatalException	This represents the updateRelationshipRule method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,RelationshipRuleDTO,relationshipRuleDTO, ,relationshipRuleDTO TODO: Explain the relationshipRuleDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	modifyRule	This service will modify the existing rule for given rule id. @param sessionContext, ruleDefinition @throws FatalException	This service will modify the existing rule for given rule id. @param sessionContext, ruleDefinition @throws FatalException
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	executeRule		This represents the checkMandatoryRelationshipForPartyType method @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,RelationshipRuleDTO,relationshipRuleDTO, ,relationshipRuleDTO TODO: Explain the relationshipRuleDTO above @return {@link DomainObjectDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyRulesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyRulesAdapter	executeRule		
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRule	Method will be used to execute rule with given rule context which contains RuleId and all other required information. @param ruleContext @return RuleExecutionResponse	This method will be used to execute rule with given Rule context.  Mandatory fields to be present in RuleContext -   <ol> <li>RuleId <li>Domain <li>DomainCategory <li>Fact information. </ol>
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRule	Method will be used to execute rule with given rule context which contains RuleId and all other required information. @param ruleContext @return RuleExecutionResponse	This method is used to evaluate the rule to check whether the account is regulated or not TODO : to check the facts need to be sent to rule from FS side

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRule	Method will be used to execute rule with given rule context which contains RuleId and all other required information. @param ruleContext @return RuleExecutionResponse	This method will be used to execute bonus interest processing.  Sequence for execution-  <ol><li>Inquire product id by loading DDAccount.<li>check for UDM maintenance at product level if present then <li>Get all the facts from consulting to evaluate the rule. <li>Validate rule for bonus interest eligibility and if account is eligible for bonus interest then <li>log the non financial statement item with description. <li>check if bonus interest is applicable for current cap cycle then call Interest module to get the start date and end date of current cycle and Call pricing to create back dated UDM Definition at account level, provide start date and end date also.  and instead of logging backdated action process backdated right away. <li>And if bonus interest is applicable for next cap cycle then create future dated UDM with start and end date and log the rate change action with due date as start date. </ol> @param applicationContext Application Context which will be set in batch process. @param actionKey @param actionRequest every subclass need to cast this to appropriate subclass of {@link ActionKey} @param dueDate due date for the action
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRule	Method will be used to execute rule with given rule context which contains RuleId and all other required information. @param ruleContext @return RuleExecutionResponse	This method will be used to execute rule.Rule module will be called to execute rule through adapter. @param ruleId @param accountId @return isBonusInterestApplicable
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRule	Method will be used to execute rule with given rule context which contains RuleId and all other required information. @param ruleContext @return RuleExecutionResponse	This method is used to execute the rule to check the account is regulated or not @param factInformationDTOs TODO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRuleForRegulatedOD		This method is used to evaluate the rule to check whether the account is regulated or not TODO : to check the facts need to be sent to rule from FS side
IDDARuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDARuleAdapter	executeRuleForRegulatedOD		This method is used to execute the rule to check the account is regulated or not @param factInformationDTOs TODO @return
ILVREventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILVREventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	This Exposure For a given Facility for LTV Cal. Maximum{(EffectiveLimit + Transfer - Utilization + Repaid(if Revolving)) , Zero} + Utilization - Repaid(if Revolving). @param facility @return Money @throws FatalException
ILVREventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILVREventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	This Exposure For a given Facility for LTV Cal. Maximum{(EffectiveLimit + Transfer - Utilization + Repaid(if Revolving)) , Zero} + Utilization - Repaid(if Revolving). @param facility @return Money @throws FatalException
ILVREventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILVREventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForNewAccount	This method will be called by Adapter services of CASA/Loan to notify Asset Classification when new account is being opened. Module to provide Account Id, Account Type, Product Code, Party Id(Primary), Facility Id. NPA module to update the main facility id if the module has given a sub-facility. The next classification date will be defaulted to current posting date. @param accountId @param accountType @param productCode @param partyId @param facilityId @param accountBranchCode TODO	This method returns the Adapter for calling the origination module
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForNewAccount	This method will be called by Adapter services of CASA/Loan to notify Asset Classification when new account is being opened. Module to provide Account Id, Account Type, Product Code, Party Id(Primary), Facility Id. NPA module to update the main facility id if the module has given a sub-facility. The next classification date will be defaulted to current posting date. @param accountId @param accountType @param productCode @param partyId @param facilityId @param accountBranchCode TODO	This method will be called by Adapter services of CASA/Loan to notify Asset Classification when new account is being opened. Module to provide Account Id, Account Type, Product Code, Party Id(Primary), Facility Id. NPA module to update the main facility id if the module has given a sub-facility. The next classification date will be defaulted to current posting date.   Since 2.2 changes in NPA to accept value date ,currently there are no callers for the method.</br> @param accountId @param accountType @param productCode @param partyId @param facilityId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForAccountUpdate	This method will be called by Adapter services of CASA/Loan to notify Asset Classification whenever the Party or the Facility linked to the account is changed. The next classification date will be defaulted to current posting date. @param accountId @param accountType @param productCode @param partyId @param facilityId	This method returns the Adapter for calling the origination module
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForAccountUpdate	This method will be called by Adapter services of CASA/Loan to notify Asset Classification whenever the Party or the Facility linked to the account is changed. The next classification date will be defaulted to current posting date. @param accountId @param accountType @param productCode @param partyId @param facilityId	This method will be called by Adapter services of CASA/Loan to notify Asset Classification whenever the Party or the Facility linked to the account is changed. The next classification date will be defaulted to current posting date.   Since 2.2 changes in NPA to accept value date ,currently there are no callers for the method.</br> @param accountId @param accountType @param productCode @param partyId @param facilityId
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param accountType	
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			accountType	
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyForArrearUpdate	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param accountType	This method is to be used to notify AssetClassification module whenever there a new arrear is raised or arrear is paid for the account.The next classification date will be defaulted to current posting date. @param accountId @param accountType
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyArrearsForFullWriteOff	This method will be called by ArrearService to inform the arrears written off to asset classification module when full writeoff of the account takes place. @param accountId @param reason1 TODO @param reason2 TODO @param arrearsByTypeDTO @param writeOffDate	This method is used to only in case of writing off of an arrear. Fully writing off of an arrear is required when the amount available for appropriation is greater than the due amount. In this case the Flag which identifies whether the arrear is paid off or not is set as Y this in a way means that the due amount is Zero . The due amount is being updated in case of partial appropriation of an arrear. <ArrearDTO> arrearListToBeUpdated <p> <span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt"></span><span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt">Data:</span> </p> <p> <span style="FONT-SIZE: 10pt">Provide The following Values:</span> </p> <p> <span style="FONT-SIZE: 10pt"> <table border="1"> <tbody> <tr> <td>Class : Ways to write of an arrear</td> <td>Arrear current write off Amount</td> <td>Due Amount</td> <td>Updated Due Amount</td> <td>Is Arrear Paid Off</td> </tr> <tr> <td>Case:1</td> <td>500AUD</td> <td>200AUD</td> <td>0</td> <td>YES</td> </tr> <tr> </tr> @param applicationContext @param eventId @param accountId @param reason1 @param reason2 @param currentPrincipalBal @param provisionDrawdownAmount @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				recoveryIneligibleAmount TODO @return
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyArrearsForFullWriteOff	This method will be called by ArrearService to inform the arrears written off to asset classification module when full writeoff of the account takes place. @param accountId @param reason1 TODO @param reason2 TODO @param arrearsByTypeDTO @param writeOffDate	Intimate previous partial writeoff amount to NPA for account closure. @param accountId
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyArrearsForFullWriteOff	This method will be called by ArrearService to inform the arrears written off to asset classification module when full writeoff of the account takes place. @param accountId @param reason1 TODO @param reason2 TODO @param arrearsByTypeDTO @param writeOffDate	This method returns the Adapter for calling the origination module
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Written off. Partial write-off is not considered for notification. @param appContext @param accountId @param taskModuleType @param status	This method returns the Adapter for calling the origination module

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountNPAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountNPAAdapter	notifyOnStatusChange	This method is to be used to notify AssetClassification module whenever the status of account changes to Written off. Partial write-off is not considered for notification. @param appContext @param accountId @param taskModuleType @param status	This method is to be used to notify AssetClassification module whenever the status of account changes to Written off. Partial write-off is not considered for notification. @param appContext @param accountId @param taskModuleType @param status
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	This method will be called to approve a workitem.   The existing workitem is loaded from the repository and is checked  to ascertain that it has not been locked by some other user.  If the workitem has been locked by some other user then   WORKITEM_ACQUIRED_BY_ANOTHER_USER exception is thrown   @param workItemId @param comment @param executableStageIds @param userId @return @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	This method will be called to reject a workItem.   The existing workItem is loaded from the repository and is checked  to ascertain that it has not been locked by some other user.  If the workItem has been locked by some other user then   WORKITEM_ACQUIRED_BY_ANOTHER_USER exception is thrown   @param workflowId @param comment @param executableStageIds @param userId @return @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	This method will be called to approve a workitem.   The existing workitem is loaded from the repository and is checked  to ascertain that it has not been locked by some other user.  If the workitem has been locked by some other user then   WORKITEM_ACQUIRED_BY_ANOTHER_USER exception is thrown   @param workItemId



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param comment @param executableStageIds @param userId @return @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	This method will be used to approve the transaction.When the transaction will get committed via 2FA delay then by using approve method from the work-list one can approve the transaction.Approve method will delegate call to this method.This method will fetch the transaction log based on the work-item id and mark the message state as READY_FOR_DISPATCH. @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	approveTransaction	This method will be use to approve the transactions after the transaction is validated for Fraud check and OAAM response is Challenge2FA @param transactionParameters @throws FatalException	
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	rejectTransaction	This method will be used to reject the transaction after the transaction is validated for Fraud check and OAAM response is Challenge2FA @throws FatalException	This method will be called to reject a workItem.  The existing workItem is loaded from the repository and is checked  to ascertain that it has not been locked by some other user.  If the workItem has been locked by some other user then  WORKITEM_ACQUIRED_BY_ANOTHER_USER exception is thrown   @param workflowId @param comment @param executableStageIds @param userId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	rejectTransaction	This method will be used to reject the transaction after the transaction is validated for Fraud check and OAAM response is Challenge2FA @throws FatalException	This method will be called to reject a workItem.   The existing workItem is loaded from the repository and is checked  to ascertain that it has not been locked by some other user.  If the workItem has been locked by some other user then   WORKITEM_ACQUIRED_BY_ANOTHER_USER exception is thrown   Otherwise this method invokes the method of the respective business service   as defined in the rejectconfiguration.properties.  @param workflowId @param comment @param executableStageIds @param userId @return @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	rejectTransaction	This method will be used to reject the transaction after the transaction is validated for Fraud check and OAAM response is Challenge2FA @throws FatalException	This method will be used to reject the transaction. When the transaction will get committed by marking 2FA delay and when one called reject method from the work-list call will delegate to this method. This method will call the reverse method of the respective service. that means it is going to reverse the the transaction which is committed. @throws FatalException
IFraudAssertionProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IFraudAssertionProcessingAdapter	rejectTransaction	This method will be used to reject the transaction after the transaction is validated for Fraud check and OAAM response is Challenge2FA @throws FatalException	
IMDMBatchIntegrationWrapperAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationWrapperAdapter	populateCustomAccountEBO	This method is used to populate Account data and publish to MDM @param accountId @param accountType @throws FatalException @throws Throwable	This method invokes MDMBatchIntegrationWrapperAdapter with account  information to publish data to MDM. @param mdmBatchIntegrationLog @throws Throwable @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IMDMBatchIntegrationWrapperAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\IMDMBatchIntegrationWrapperAdapter	populateCustomPartyEBO	This method is used to populate party data and publish to MDM. @param partyId @param partyType @throws FatalException	Invokes MDMBatchIntegrationWrapperAdapter with party information to publish data to MDM. @param mdmBatchIntegrationLog @throws Throwable
IAccountingEntryDomainCatExtractionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingEntryDomainCatExtractionAdapter	extractDomainCategory		
IntegrationAnalyticsPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\analytics\IntegrationAnalyticsPollerAdapter	initiateAnalyticsPoller	This method will initiate Analytics poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	This method will initiate Analytics poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException
IntegrationAnalyticsPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\analytics\IntegrationAnalyticsPollerAdapter	initiateAnalyticsPoller	This method will initiate Analytics poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller, delayed transaction poller, FTP webservice poller. Please add more in this method. Additionally, we also have an extension for consulting pollers @param sessionContext
IAccountInquiryChannelAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IAccountInquiryChannelAdapter	fetchDetailsUsingChannel	The following method is used to fetch NamedPartyRelationshipDTO using identifier. @param identifier @return @throws FatalException	The following method is used to fetch Party and account related information based on the identifier that has been supplied. @param businessContext @param identifier @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentPartyAdapter	setPostalAddressDetails	Sets the postal address details.	Gets the delivery details for party.for each passed recipients delivery details are fetched from party. @param sessionContext @param documentDetailDTO @param documentPolicyDetailDTO @return @throws FatalException
IDocumentPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentPartyAdapter	setPostalAddressDetails	Sets the postal address details.	TODO Gets the delivery details for associate parties. @param sessionContext the session context @param documentDetailDTO the document detail dto @return the delivery details for associate parties
IDocumentPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentPartyAdapter	setPostalAddressDetails	Sets the postal address details.	This method call the party module to get party level preferences for document delivery.This method will take the relation maintain after execution of the policy in documentTypeRelationshipDTO.this DTO actually contains the name of the relationship such as Applicant/co-applicant/Guaranter. Then for each relationship define for policy level and inside that all the relationship passed by the caller check which are equal then for found relationships find out the parties pssed by the caller.For each party fetch the address details from PartyAddressApplicationService and for Email and and fax call contact application service. @param sessionContext the session context @param documentDetailDTO the document detail dto @param documentPolicyDetailDTO the document policy detail dto @return DocumentDeliveryPreferencesDTO @throws com.ofss.fc.infra.exception.FatalException @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentPartyAdapter	getPartyTypeForParty	This private method will return the { <a href="#">PartyType</a> } for the partyId passed. @param sessionContext @param partyId @return @throws FatalException	Sets the postal address details. Gets the postal details as communication details for party. @param sessionContext the session context @param documentDeliveryPreferenceDTO the document delivery preference dto @throws FatalException the fatal exception
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchLedgerDetails	This method will fetch the ledger details and returns ProductLedgerDTO @param sessionContext @param LedgerCode @return	<p><b>addJournalMultiEntry():</b> This method will be used to create a new Multi Entry into a particular batch code that has been supplied along with the input DTO. Only Direct posting category of Ledgers , as well as CASA accounts can be posted to using this service. Upon successful creation of an entry , a record will be created in the table</p> <p><i>FLX_JE_MULTENT_MAIN_B</i> and <i>FLX_JE_MULTENT_OFFSET_B</i> , as well as in the accounting entry data stores, with the data supplied The data will be subject to many validations before a successful save. Part of the validations will be done within the Journal Entry(JE) module and the rest will be within the Accounting module This service will make calls to the CASA service as well if the transaction account being posted is a CASA account The call to CASA service will be made to update the balance, but the entry will be passed in accounting , with a subsequent call to the accounting service. There are no product templates associated with an entry originating from within the JE module There will be a call to update the Batch Totals in the Batch Record of the Batch Code being used . The columns that will be updated are , <i>DR_TOTAL_AMT</i> , <i>CR_TOTAL_AMT</i> and <i>LOCKED_FLAG</i> in the table <i>FLX_JE_BATCH_CODES_ALL_B</i> Locked Flag will be set to 'Y'. This will call <code>JournalMultiEntryValidationsAdd</code> for the Validations ,</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				which includes the not null validations as well as theValidations that requires calls to other modules through IMC @fcb.param MI,SessionContext,sessionContext, ,holds the basic information about the user of the system. @fcb.param MI,JournalMultiEntryAppDTO,journalMultiEntryDT O, ,holds data transfer object for the  Journal Multi Entry. @return {@link com.ofss.fc.app.je.multipleentry.dto.JournalMultiEntry InquiryResponse} An instance   of JournalMultiEntryInquiryResponse contains the generated identifier for the Multi entry.   @throws FatalException
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchLedgerDetails	This method will fetch the ledger details and returns ProductLedgerDTO @param sessionContext @param LedgerCode @return	This method is used to fetch the Account and Ledger flag @fcb.param MI,SessionContext,sessionContext, ,holds the basic information about the user of the system. @fcb.param MI,JournalMultiEntryAppOffset,journalMultiEntryAppOffset, ,holds the array of Offset Details  @return {@link com.ofss.fc.app.je.multipleentry.dto.JournalMultiEntryAppOffset} Array of Offset Details   with the account or Ledger Flag
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchLedgerDetails	This method will fetch the ledger details and returns ProductLedgerDTO @param sessionContext @param LedgerCode @return	validateData (validate multi entry input data and call to validate Gaap and Ledger through adapter)
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchLedgerDetails	This method will fetch the ledger details and returns ProductLedgerDTO @param sessionContext @param LedgerCode @return	This method does the Batch Balancing By passing Suspense Entries for Un balanced Batches @param sessionContext @param journalEntryBatch @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchLedgerDetails	This method will fetch the ledger details and returns ProductLedgerDTO @param sessionContext @param LedgerCode @return	 This method validates ledger code entered with the ledger code   of the Accounting module through an IMC Adapter. If ledger code is not   maintained by Product Ledger maintenance then this method raises a validation error.   If the ledger code entered is maintained and its posting type is <b>INDIRECT</b> then this method   raises a validation error. 
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidGaapCode	@param sessionContext @param GaapCode @return boolean	
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidGaapCode	@param sessionContext @param GaapCode @return boolean	validateData (validate multi entry input data and call to validate Gaap and Ledger through adapter)
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidGaapCode	@param sessionContext @param GaapCode @return boolean	This validate the Gaap Code
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidLedgerCode	@param sessionContext @param ledgerCode @return boolean	validateData (validate multi entry input data and call to validate Gaap and Ledger through adapter)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidLedgerCode	@param sessionContext @param ledgerCode @return boolean	This validate the Transaction Account Code
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	isValidLedgerCode	@param sessionContext @param ledgerCode @return boolean	 This method validates ledger code entered with the ledger code   of the Accounting module through an IMC Adapter. If ledger code is not   maintained by Product Ledger maintenance then this method raises a validation error.   If the ledger code entered is maintained and its posting type is <b>INDIRECT</b> then this method   raises a validation error. 
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchBranchParameter	@param sessionContext @param bankCode @param BranchCode @return BranchParameterDTO	
IJournalProductLedgerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\je\IJournalProductLedgerAdapter	fetchBranchParameter	@param sessionContext @param bankCode @param BranchCode @return BranchParameterDTO	This method returns existing JournalSingleEntryDTO for a batch @param response @param journalEntryBatch @param sessionContext @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	acquireFundsFromTermDeposit	This method will acquire funds required by sweep in beneficiary. A lien will be placed on TD account for the amount which is acquired   by the beneficiary account. This method is invoked when a TD account is maintained as a SweepIn provider for a CASA beneficiary account.  And the CASA beneficiary account is short of funds to honour the debit initiated by the customer.  @param beneficiaryAccountId @param providerAccountId @param requireFunds @param requestDate @return	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	acquireFundsFromTermDeposit	This method will acquire funds required by sweep in beneficiary. A lien will be placed on TD account for the amount which is acquired   by the beneficiary account. This method is invoked when a TD account is maintained as a SweepIn provider for a CASA beneficiary account.  And the CASA beneficiary account is short of funds to honour the debit initiated by the customer.  @param beneficiaryAccountId @param providerAccountId @param requireFunds @param requestDate @return	This method calls TD service to acquire funds from a Term Deposit Account via adapter. @param providerAccountId @param fundsToRetrieve @param instructionRequest

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	isValidAccountForSweepIn	This method validates term deposit account, whether it is valid Term Deposit account for SweepIn Maintenance. @param accountId @param accountCurrency @return	This method checks the validity of the provider account.
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	isValidAccountForSweepIn	This method validates term deposit account, whether it is valid Term Deposit account for SweepIn Maintenance. @param accountId @param accountCurrency @return	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	isValidAccountForSweepIn	This method validates term deposit account, whether it is valid Term Deposit account for SweepIn Maintenance. @param accountId @param accountCurrency @return	<ul style="list-style-type: none"> <li>&lt;ol&gt; Based on module type the corresponding account gets loaded.If module type = DDA then</li> <li>&lt;li&gt;create instance account of {@link DemandDepositAccount}&lt;/li&gt;</li> <li>&lt;li&gt;set the {@link DemandDepositAccountKey} with AccountTransaction.accountId&lt;/li&gt;</li> <li>&lt;li&gt;load DemandDepositAccount by {@link DemandDepositAccountRepository#load(demandDepositAccount: DemandDepositAccount)}&lt;/li&gt;</li> <li>&lt;li&gt;return account instance&lt;/li&gt;</li> </ul>
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	removeLienFromTermDeposit	This method will reinstate the lien placed on a Term Deposit account. It might possible that the lien amount is fully recovered or partially recovered.  @param beneficiaryAccountId @param providerAccountId @param fundsToReinstateLien @param requestDate @return	This method is the abstract method for funding an account. The implementation will change based on the type of account being funded. Each concrete class is supposed to provide the correct implementation of this method based on the type of account that it is responsible for. For instance, the DemandDepositFundingStrategy will provide the implementation for funding a demand deposit account. @param beneficiaryAccountId @param fundsToDistribute

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccountForSweepIn	This method will fetch product Id of a TD Account for validation of product combination for SweepIn instruction. @param accountId @return @throws FatalException	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccountForSweepIn	This method will fetch product Id of a TD Account for validation of product combination for SweepIn instruction. @param accountId @return @throws FatalException	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchPartyIdOfTermDepositAccount	This method is used to determine the party Id of a Term Deposit Account. @param accountId @return @throws FatalException	This method check whether transfer between these accounts is allowed or not. If toAccount is null then it is treated as external account. @param fromAccount @param fromAccountType @param toAccount @param toAccountType @return
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchPartyIdOfTermDepositAccount	This method is used to determine the party Id of a Term Deposit Account. @param accountId @return @throws FatalException	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for TD principal payout where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for TD maturity where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Loan disbursement where the account number passed is beneficiary account number in Loan disbursement setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Sweep Out where the account number passed is beneficiary account number/provider account number in Sweep Out setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount
IDDATDAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDATDAdapter	fetchAccount	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	<p>&lt;ol&gt;The method populates the DepositsFeeRequestInfo which is then passed to the FeesManagementService as input. The FeeManagementService requires instance of DepositsFeeRequestInfo &lt;/ol&gt;</p> <p>&lt;ol&gt;The method set the TaskModuleType as follows: &lt;li&gt;if the moduleType == DDA then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.CASA&lt;/li&gt; &lt;li&gt;if the moduleType == TD then set DepositsFeeRequestInfo.ModuleType = TaskModuleType.TERM_DEPOSIT&lt;/li&gt; &lt;li&gt;load account by @link fetchDemandDepositAccount(accountId: String, moduleType: ModuleType): IAccount&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.partyId = account.getPartyId()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.accountId = accountTransaction.getAccountId()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.eventId = accountTransaction.getEventId()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.currencyCode[0] = accountTransaction.getCurrency()&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.PricingDetail = @link getPricingDetail(FeeDetails: feeDetails): PricingDetails&lt;/li&gt; &lt;li&gt;set depositsFeeRequestInfo.EvaluationDate =</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				AccountTransaction.TransactionDates.PostingDate</li> > <li>set depositsFeeRequestInfo.userReferenceNumber = accountTransaction.UserReferenceNumber</li> <li>set depositsFeeRequestInfo.TransactionAmount = accountTransaction.TransactionAmount.AmountInAc countCurrency</li </ol> @param accountTransaction @param moduleType @param feeAmounts @param priceTicketNumber @return DepositsFeeRequestInfo
IDDATDAda pter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dda\IDDATDAdapter	fetchAccou nt	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	
IDDATDAda pter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dda\IDDATDAdapter	fetchAccou nt	This method will provide the from offer code for given account details @param accountId @return ITermDepositAccount	<ol> Based on module type the corresponding account gets loaded.If module type = DDA then <li>create instance account of {@link DemandDepositAccount}</li> <li>set the {@link DemandDepositAccountKey} with AccountTransaction.accountId</li> <li>load DemandDepositAccount by {@link DemandDepositAccountRepository#load(demandDep ositAccount: DemandDepositAccount)}</li> <li>return account instance</li> </ol> @param accountId @param moduleType
ISettlementC ollectionAda pter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \settlement\ISettlement CollectionAdapter	modifyUsin gCounterpa rtyDetails	The following method is used to modify the counterparty details. @param objectValue @param partyId @throws FatalException	The following method is used to modify the component Preference based on the counter party that is being supplied. @param componentPreferenceDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementCollectionAdapter	createCounterparty	The following method is used to create new counter party id. @param objectValue @param partyId @throws FatalException	
ISettlementCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementCollectionAdapter	createCounterparty	The following method is used to create new counter party id. @param objectValue @param partyId @throws FatalException	This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction INDicates the direction of fund flow from Customer perspective @param transactionType INDicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled
ISettlementCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementCollectionAdapter	createCounterparty	The following method is used to create new counter party id. @param objectValue @param partyId @throws FatalException	The following method is called to modify the response @param detailsDTO @param counterpartyInquiryResponse @throws FatalException
ISettlementCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementCollectionAdapter	validateCounterparty	The following method is used to create new counter party id. @param objectValue @param partyId @throws FatalException	This method is used to validate the inter product transfer.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		Populate the application context object form batchContextObject. @param context
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		Fetches financial dates namely posting date and value date depending upon the parameter set in the flexj.properties file.
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} @param context @param bankCode @return BranchDatesDefinitionDTO @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the posting date of the current branch and bank. It will fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} Check for the CutoffRunToday() This method will return the maximum of the two, process date and posting date. @param branchCode @return Date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the posting date of the current branch. If branchCode is equal to applicationContext.getTransactionBranchCode() return applicationContext.getPostingDate().Otherwise, fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} Check for the CutoffRunToday() This method will return the maximum of the two, process date and proasting date. @param branchCode @return Date
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the process date of the current branch. If branchCode is equal to applicationContext.getTransactionBranchCode() return applicationContext.getProcessDate().Otherwise, fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} and return branchDates.getCurrentDate(). @param branchCode @return Date
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		@modelguid {F9A807CA-230C-4B75-B86E-D5A8BB7B195F}
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		Populate the application context object form batchContextObject. @param context
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or ifThis method is called before any

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				other method in this service @author: SureshRag
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan .  This should be used <b><em>while UDM addition is not done by ELPBC screen</em></b>. @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or if this method is called before any other method in this service @author: SudipG
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		Populate the application context object form batchContextObject. @param context
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		Fetches financial dates namely posting date and value date depending upon the parameter set in the flexj.properties file.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} @param context @param bankCode @return BranchDatesDefinitionDTO @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the posting date of the current branch and bank. It will fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} Check for the CutoffRunToday() This method will return the maximum of the two, process date and posting date. @param branchCode @return Date
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the posting date of the current branch. If branchCode is equal to applicationContext.getTransactionBranchCode() return applicationContext.getPostingDate().Otherwise, fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} Check for the CutoffRunToday() This method will return the maximum of the two, process date and posting date. @param branchCode @return Date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		This method will used to fetch the process date of the current branch. If branchCode is equal to applicationContext.getTransactionBranchCode() return applicationContext.getProcessDate().Otherwise, fetch the branch dates from core service. Call {@link com.ofss.fc.app.adapter.IAdapterFactory.getAdapter(String)} to get the adapter for Core Service and give a call {@link com.ofss.fc.app.adapter.core.ICoreApplicationServiceAdapter.fetchBranchDates(SessionContext, String, String)} and return branchDates.getCurrentDate(). @param branchCode @return Date
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		@modelguid {F9A807CA-230C-4B75-B86E-D5A8BB7B195F}
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchDates		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		Populate the application context object form batchContextObject. @param context
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or ifThis method is called before any other method in this service @author: SureshRag
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This overloaded method will be called after account or deposit is opened for   adding rate variances to the pricing plan .  This should be used <b><em>while UDM addition is not done by ELPBC screen</em></b>. @param arrangementNumber the arrangement/account number with which to associate the pricing plan @param domain : domain name @param domainCategory : domain category name like Loan @param eventRateVariances : list of EventRateVariance {@link InformationObject} that encapsulates the rate variance information for a given event. @throws FatalException if the pricing ticket is invalid or the pricing contract service throws a {@link FatalException} or ifThis method is called before any other method in this service @author: SudipG

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @param transactionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		This method will createApplicationContext from the SessionContext and the TransactionContext. The fields like TransactionReferenceNumber, BatchNumber will be fetched from the TransactionContext. and other fields such as the posting date and transactionDate will be fetch from the sessionContext.This method is mainly used for calling the domain services from the managers. @param sessionContext @return ApplicationContext @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Dates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranch Fields		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchFields		@param sessionContext @param branchId @return
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBranchFields		Populate Static Parameters with missing attributes @param sessionContext @param reportStaticParam @return @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		@param sessionContext @return
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		Populate Static Parameters with missing attributes @param sessionContext @param reportStaticParam @return @throws FatalException
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankDates		This method will create Rule Set. @param ruleSetDTO DTO containing rule set details @throws FatalException <ul> Error Reason may be one of the following <li> {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULESET_EFFECTIVE_DATE}</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankDates		This method will modify the rule set. If rule set is not yet effective then it will override old rule set definition. Otherwise will store this as new rule set. In order to store this as new rule set effective date must be different than old rule set @param ruleSetId id of rule set being modified @param effectiveDate Effective date of Rule being modified @param modifiedRuleSetDTO DTO containing modified rule set details @throws FatalException <ul> Error Reason may be one of the following <li> {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_PAST_RULESET_EFFECTIVE_DATE}</li> <li> {@link com.ofss.fc.domain.rule.constants.RuleErrorConstants#MID_RULESET_EFFECTIVE_TODAY}</li> </ul>
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankDates		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		@param sessionContext @return
ICoreApplicationServiceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\core\ICoreApplicationServiceAdapter	fetchBankFields		Populate Static Parameters with missing attributes @param sessionContext @param reportStaticParam @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	loadParty	@param partyId @return @throws FatalException	The following method is used to set the party in context. Party id is provided by the user in the conversation command.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	loadParty	@param partyId @return @throws FatalException	The following method is used to validate party name and set details. @param contextResponse specifies the response class. @param partyAccountDetailsChangedDTO specifies PartyAccountDetailsChangedDTO. @param partyName specifies party name. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	loadParty	@param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	loadParty	@param partyId @return @throws FatalException	The following method is used to set the application in context. Application id is provided by the user in the conversation command.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingNickName	The following method is used to fetch details using nickname @param partyId @param nickName @return @throws FatalException	The following method is used to fetch Party and account related information based in the identifier that has been supplied. @param businessContext @param identifier @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	The following method is used to fetch party details using nickname
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchPartyAccountRelationship	The following method is used to fetch party details using nickname	The following emthdo is used to fetch the party account relationship; @param partyId @param nickName @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	createContactPoint	The following method is used to create contact point type @param sessionContext @param dto @return @throws FatalException	The following method is used to change the contact point details.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	createContactPoint	The following method is used to create contact point type @param sessionContext @param dto @return @throws FatalException	The following method is used to create contact points @param contactPointDTO @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	createContactPoint	The following method is used to create contact point type @param sessionContext @param dto @return @throws FatalException	The following method is used to validate record not found exception. @param partyAdapter specifies the party adapter. @param contactPointDTO specifies the contactPointDTO. @param contactPointType specifies contactPointType. @param contactPreferenceType specifies contactPreferenceType. @param contactPointTypeBundle specifies contactPointType bundle. @param contactPreferenceTypeBundle specifies contactPreferenceType bundle. @param e specifies the exception. @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	updateContactPoint	The following method is used to update the contact point dto @param dto @return @throws FatalException	The following method is used to change the contact point details.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	updateContactPoint	The following method is used to update the contact point dto @param dto @return @throws FatalException	The following method is used to update contact points @param contactPointDTO @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	deleteContactPoint	The following method is used to delete the contact point type. @param dto @return @throws FatalException	The following class is used to execute the Delete contact point details belonging to the party in context.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchContactPoint	The following method is used to fetch the contact points @param sessionContext @param dto @return @throws FatalException	The following method is used to change the contact point details.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchContactPoint	The following method is used to fetch the contact points @param sessionContext @param dto @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchContactPoint	The following method is used to fetch the contact points @param sessionContext @param dto @return @throws FatalException	The following method is used to set the account in context. The following method is used to execute conversation text depending on the type of the transaction. This is a common method exposed on various channels to execute conversation text. Parameters are common across all the channels. The EObject is used to extract a command object which is used to perform operation. @param etask @param contextResponse @param businessContextDTO @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllContactPointsByParty	The following method will be used to fetch all @param sessionContext @param partyId @return @throws FatalException	The following method is used to execute the fetch the registered mobile numbers based on the party id . The information is stored based on the Contact point Type.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllContactPointsByParty	The following method will be used to fetch all @param sessionContext @param partyId @return @throws FatalException	The following method is used to execute the fetch the registered land line numbers based on the party id . The information is stored based on the Contact point Type.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllContactPointsByParty	The following method will be used to fetch all @param sessionContext @param partyId @return @throws FatalException	The following method is used to fetch all the registered contact points which are related to the party in context.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllContactPointsByParty	The following method will be used to fetch all @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllContactPointsByParty	The following method will be used to fetch all @param sessionContext @param partyId @return @throws FatalException	The following method is used to execute the fetch the registered email based on the party id .The information is stored based on the Contact point Type.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireByType	The following method is used to inquire all the nicknames on the basis of party id and relationship type @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireByType	The following method is used to inquire all the nicknames on the basis of party id and relationship type @param partyId @return @throws FatalException	The following method is used to execute the fetch the registered linked accounts based on the party id . The information is stored based on the Contact point Type.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireByType	The following method is used to inquire all the nicknames on the basis of party id and relationship type @param partyId @return @throws FatalException	The following method is used to execute the fetch the registered billers based on the party id .The information is stored based on the Contact point Type.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	The following method is used to fill NamedPartyRelationship map. @param partyId specifies the party id. @param namedPartyRelationshipInquiryResponse specifies NamedPartyRelationshipInquiryResponse @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	The following method is used to fetch NamedPartyRelationshipDTO using identifier. @param identifier @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireAll	The following method is used to inquire all nick names using the party id. @param sessionContext @param partyId @return @throws FatalException	The following method is used to fetch NamedPartyRelationshipDTO using identifier. @param identifier @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchDetailsUsingMaskedIdentifier	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	The following method is used to fetch Party and account related information based in the identifier that has been supplied. @param businessContext @param identifier @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireByProvidingRelationshipType	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	The following method is used to execute conversation text depending on the type of the transaction. This is a common method exposed on various channels to execute conversation text. Parameters are common across all the channels. The EObject is used to extract a command object which is used to perform operation. The following adapter is used to set the account in context. @param etask @param contextResponse @param businessContextDTO @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	inquireByProvidingRelationshipType	The following method is used to fetch nick name related details using the identifier. @param partyId @param identifier @return @throws FatalException	The following method is used to fetch Party and account related information based in the identifier that has been supplied. @param businessContext @param identifier @return @throws FatalException
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchSummaryDetailsForAccount	@param sessionContext @param partyAccountDTO @return @throws FatalException	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	getLoanAccountBalanceDetails	This method takes an accountId and fetches account and balance info for a given loan account @param sessionContext   @param accountId   @return PartyAccountBalancesInquiryResponse  @throws FatalException  	The following method is used to fetch the balances for the account which is in context. Balance display pattern is different for different account type's.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	getLoanAccountBalanceDetails	This method takes an accountId and fetches account and balance info for a given loan account @param sessionContext   @param accountId   @return PartyAccountBalancesInquiryResponse  @throws FatalException  	
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllDeposits	This method takes an accountId and fetches all deposits for the given term deposit account @param sessionContext   @param accountId   @return DepositResponse  @throws FatalException  	The following method is used to fetch the balances for the account which is in context. Ballance display pattern is different for different account type's.
IDSLPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLPartyAdapter	fetchAllDeposits	This method takes an accountId and fetches all deposits for the given term deposit account @param sessionContext   @param accountId   @return DepositResponse  @throws FatalException  	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionFeeRecoverySequence	<p>This method will fetch <a href="#">FeeRecoverySequence</a> to be followed from Product Manufacturing Unit module.</p> <p>This <a href="#">FeeRecoverySequence</a> will be used to apply redemption penalty charges during pre-mature redemption. @param productId The Product for which <a href="#">FeeRecoverySequence</a> needs to be fetched @return <a href="#">FeeRecoverySequence</a> for given product id</p>	<p>This method is the Starting Point for Maturity Processing. It is Invoked from <a href="#">DepositMaturityActionExecutor#executeAction</a>. At a Broad level the method performs the following actions</p> <ul style="list-style-type: none"> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke <a href="#">DepositMaturityService#processDepositMaturity</a>. This method returns a <a href="#">MaturityResponse</a> object. The Response object contains the following Information <ul style="list-style-type: none"> <li>The <a href="#">RenewalResponse</a> containing any amount which has to be renewed</li> <li>The <a href="#">MaturityPayoutResponse</a> contains Principal, Interest Amount which is to be paid out and the <a href="#">PayoutInstruction</a></li> <li>The Deposit which is being Matured</li> </ul> </li> <li>The method reads the <a href="#">RenewalResponse</a>. If Not null then it invokes <a href="#">DepositRenewalService#autoRenewDeposit</a>. If New Deposit received from <a href="#">DepositRenewalService</a> is not Null then <a href="#">DepositMaturityService#processPostMaturityRenewal</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</li> <li>If Deposit Returned from Renewal is null then <a href="#">DepositMaturityService#matureDeposit</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured</li> <li>If PayoutResponse object is present then the method determines the type of Payout. If Principal Payout Instruction is Maintained then <a href="#">DepositMaturityService#processPostMaturityRenewal</a> is invoked.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionFeeRecoverySequence	<p>This method will fetch {@link FeeRecoverySequence} to be followed from Product Manufacturing Unit module.&lt;br&gt; This {@link FeeRecoverySequence} will be used to apply redemption penalty charges during pre-mature redemption. @param productId The Product for which {@link FeeRecoverySequence} needs to be fetched @return {@link FeeRecoverySequence} for given product id</p>	<p>This method fetches {@link FeeRecoverySequence} from PMU module. @param productId TD product id @return {@link FeeRecoverySequence} The sequence for fee recovery</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionFeeRecoverySequence	This method will fetch {@link FeeRecoverySequence} to be followed from Product Manufacturing Unit module.  This {@link FeeRecoverySequence} will be used to apply redemption penalty charges during pre-mature redemption. @param productId The Product for which {@link FeeRecoverySequence} needs to be fetched @return {@link FeeRecoverySequence} for given product id	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getRedemptionFeeRecoverySequence(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPrematureRedemptionPenaltyRoutine	This method will fetch {@link PenaltyRoutineType} to be followed from Product Manufacturing Unit module.  This {@link PenaltyRoutineType} will be used to decide premature redemption penalty routine @param productId The Product for which {@link PenaltyRoutineType} needs to be fetched @return {@link PenaltyRoutineType} for given product id	This method will derive the strategy id to be used for Term Deposit Redemption  @param redemptionType {@link RedemptionType} being used for Term Deposit Redemption @return Identifier of Strategy to be used for Term Deposit Redemption.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPrematureRedemptionPenaltyRoutine	This method will fetch {@link PenaltyRoutineType} to be followed from from Product Manufacturing Unit module.  This {@link PenaltyRoutineType} will be used to decide premature redemption penalty routine @param productId The Product for which {@link PenaltyRoutineType} needs to be fetched @return {@link PenaltyRoutineType} for given product id	This method returns calculated economic cost @param deposit Term deposit details @param redemptionDate Date on which redemption done @param redemptionAmount Amount to be redeemed @return Money Penalty amount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPrematureRedemptionPenaltyRoutine	This method will fetch {@link PenaltyRoutineType} to be followed from from Product Manufacturing Unit module.  This {@link PenaltyRoutineType} will be used to decide premature redemption penalty routine @param productId The Product for which {@link PenaltyRoutineType} needs to be fetched @return {@link PenaltyRoutineType} for given product id	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getPrematureRedemptionPenaltyRoutine(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumInterestEligibility	This method will fetch minimum term interest eligibility.   No interest will be paid for pre-mature full redemption if deposit is in minimum term interest eligibility  @param productId The Product for which minimum term interest eligibility needs to be fetched @param currencyCode The currency allowed for the given product @return int minimum term	This method returns expiry Date for minimum Term for Interest eligibility for this deposit. @return Expiry Date for minimum Term for Interest eligibility for this deposit.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			interest eligibility	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinTermIntEligibility	This method will fetch minimum term interest eligibility.   No interest will be paid for pre-mature full redemption if deposit is in minimum term interest eligibility  @param productId The Product for which minimum term interest eligibility needs to be fetched @param currencyCode The currency allowed for the given product @return int minimum term interest eligibility	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMinTermIntEligibility(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount.  @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method ensures that the principalBalance is in range of amount defines at product level
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount.  @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method validates the redemption amount <ul><li>whether its less than principal amount</li><li>whether the balance is less than minimum deposit amount after part redemption</li> <li>whether the redemption amount is in multiple of redemption units</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method validates initial investment amount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method fetchs the Interest Details. This output DTO{@link TDInterestSimulationInterestRateDTO} contains {@link FactsRequiredForSimulationResponse} which stores the facts required for Simulation process, minimum deposit amount and minimum term specified for product, deposit frequency details specified at product level. @param sessionContext @param productId @param partyId @param currencyCode @return TDInterestSimulationInterestRateDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method fetches the Interest Rate for the given product from pricing module. @param sessionContext @param simulationParametersDTO @param termDepositFactsDTO @return interestRateFromPricing @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product	This method validates initial investment amount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@return BigDecimal minimum payin amount	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount.  @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount.  @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method validates the redemption amount <ul> <li>whether its less than principal amount</li> <li>whether the balance is less than minimum deposit amount after part redemption</li> <li>whether the redemption amount is in multiple of redemption units</li> </ul>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount.  @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	If the new deposit amount does not satisfy the minimum amount for the TD as set at the Product level  Throw message as " Modification of the contract not permitted. Deposit does not satisfy the minimum deposit amount" <p> If the new deposit amount does not satisfy the maximum amount for the TD as set at the Product level   Throw message as "Modification of the contract not permitted. Deposit does not satisfy the maximum deposit amount criteria" @param newPrincipalAmount {@link Balance#getPrincipleBalance() }

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	<p>This method return product minimum payin amount.  @<a href="#">param productId</a> The product for which minimum payin amount needs to be fetched  @<a href="#">param currencyCode</a> The currency allowed for the given product  @<a href="#">return BigDecimal</a> minimum payin amount</p>	<p>This method is the Starting Point for Maturity Processing. It is Invoked from <a href="#">{@link DepositMaturityActionExecutor#executeAction}</a>. At a Broad level the method performs the following actions</p> <ol style="list-style-type: none"> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke <a href="#">{@link DepositMaturityService#processDepositMaturity}</a>. This method returns a <a href="#">{@link MaturityResponse}</a> object. The Response object contains the following Information <ol style="list-style-type: none"> <li>The <a href="#">{@link RenewalResponse}</a> containing any amount which has to be renewed</li> <li>The <a href="#">{@link MaturityPayoutResponse}</a> contains Principal, Interest Amount which is to be paid out and the <a href="#">{@link PayoutInstruction}</a></li> <li>The Deposit which is being Matured</li> </ol> </li> </ol> <p>The method reads the <a href="#">RenewalResponse</a>.  If Not null then it invokes <a href="#">{@link DepositRenewalService#autoRenewDeposit}</a>. If New Deposit received from <a href="#">DepositRenewalService</a> is not Null then <a href="#">{@link DepositMaturityService#processPostMaturityRenewal}</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</p> <p>If Deposit Returned from <a href="#">Renewal</a> is null then <a href="#">{@link DepositMaturityService#matureDeposit}</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured</p> <p>If <a href="#">PayoutResponse</a> object is present then the method determines the type of Payout.  If <a href="#">Principal Payout Instruction</a> is Maintained then <a href="#">{@link</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				#doPrincipalPayout} is invoked. </p> <p> If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked </p> In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked</li> </ol> Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method validates the Redemption amount with the product minimum balance maintained at product level @param redemptionAmount @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method checks whether the total redemption amount of premature part redemption instructions is greater than principal balance of deposit. @param payoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method ensures that the principalBalance is in range of amount defines at product level
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	This method do the inquiry of redemption amount. @param deposit Term deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinDepositAmount	This method return product minimum payin amount. @param productId The product for which minimum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal minimum payin amount	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMinDepositAmount(java.lang.String, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method ensures that the principalBalance is in range of amount defines at product level
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method validates initial investment amount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum	This method validates initial investment amount



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			payin amount	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	If the new deposit amount does not satisfy the minimum amount for the TD as set at the Product level  Throw message as " Modification of the contract not permitted. Deposit does not satisfy the minimum deposit amount" <p> If the new deposit amount does not satisfy the maximum amount for the TD as set at the Product level   Throw message as ⌈⌋ Modification of the contract not permitted. Deposit does not satisfy the maximum deposit amount criteria⌈⌋ @param newPrincipalAmount {@link Balance#getPrincipleBalance()}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances </p> <p> If Deposit Returned from Renewal is null then {@link DepositMaturityService#matureDeposit} is invoked, which will update the Deposit Balances and mark the Deposit as Matured </p> </li> <li>If PayoutResponse object is present then the method determines the type of Payout. <p> If Principal Payout Instruction is Maintained then {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method ensures that the principalBalance is in range of amount defines at product level
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaxDepositAmount	This method return product maximum payin amount. @param productId The product for which maximum payin amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal maximum payin amount	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMaxDepositAmount(java.lang.String, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsPrematureRedemptionAllowed	This method returns true if premature redemption is allowed for the product, else false. @param productId The product for which flag IsPrematureRedemptionAllowed needs to be fetched @return True if premature redemption allowed, else false	This method validates the product preference whether premature redemption is allowed or not at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsPrematureRedemptionAllowed	This method returns true if premature redemption is allowed for the product, else false. @param productId The product for which flag IsPrematureRedemptionAllowed needs to be fetched @return True if premature redemption allowed, else false	This method validates the product preference whether premature redemption is allowed or not at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsPrematureRedemptionAllowed	This method returns true if premature redemption is allowed for the product, else false. @param productId The product for which flag IsPrematureRedemptionAllowed needs to be fetched @return True if premature redemption allowed, else	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsPrematureRedemptionAllowed(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			false	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnltyFullRedem	This method returns true if it is allowed to waive penalty at full redemption  @param productId The product for which flag IsWaivePnltyFullRedem needs to be fetched @return True if waiving penalty at full redemption allowed, else false.	This method do the inquiry of redemption amount. @param deposit Term deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnltyFullRedem	This method returns true if it is allowed to waive penalty at full redemption  @param productId The product for which flag IsWaivePnltyFullRedem needs to be fetched @return True if waiving penalty at full redemption allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsWaivePnltyFullRedem(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnltyPartRedem	This method returns true if it is allowed to waive penalty at part redemption  @param productId The product for which flag IsWaivePnltyPartRedem needs to be fetched @return True if waiving penalty at part redemption allowed, else false.	This method do the inquiry of redemption amount. @param deposit Term deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnltyPartRedem	This method returns true if it is allowed to waive penalty at part redemption  @param productId The product for which flag IsWaivePnltyPartRedem needs to be fetched @return True if waiving penalty at part redemption allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsWaivePnltyPartRedem(String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the lien flag.  This field can not have false in case any lien exists on the deposit.  If at the product level lienAllowed is false then this field has true then throw exception.  @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	@return the isLienAllowed
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	@param isLienAllowed the isLienAllowed to set
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the Lien amount Entered and fills the following fields in the LienAmountDTO <ul> <li>Lien Start Date : Defaulted to Posting Date</li> <li>Lien Expiry Date : Defaulted to Deposit Maturity Date</li> <li>Lien Principal Amount : principalBalance-LienedPrincipal if((principalBalance-LienedPrincipal)<lienAmount)</li> <li>Lien Interest Amount :lienAmount- Lien Principal Amount</li> <li>Principal Balance of the deposit</li> <li>Compounded Interest maintained if any</li></ul> @param accountId @param depositId @return LienDetailsResponse
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param termDepositAccountProductAdapter @param deposit @param productId @param lienAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This methods sets the LienPrincipal and LienInterest amount to Lien Details @param lienDetails @param updateFlag
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the business rules for the lien placing @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw Invalid amount entered. Please enter an amount greater than Zero. @param lienDetails



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message "Account Status is closed and lien cannot be marked". <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit".  @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the business rules for the lien placing @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				⌋ Invalid amount entered. Please enter an amount greater than Zero⌋. @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message ⌋ Account Status is closed and lien cannot be marked ⌋. <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit⌋.  @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the business rules for the lien placing @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message <math>\infty</math> Account Status is closed and lien cannot be marked <math>\infty</math>. <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit<math>\infty</math>.  @return boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	This method validates the business rules for the lien placing @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product. @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message "Account Status is closed and lien cannot be marked". <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed /closed status and lien cannot be marked on the deposit".  @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowed(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowedOnPrincipal(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowed	This method return if the lien is allowed at the product.  @param productId The product for which LienAllowed needs to be fetched @return True if lien is allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowedOnPrincipalAndInterest(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level. @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	This method validates the Lien amount Entered and fills the following fields in the LienAmountDTO <ul style="list-style-type: none"> <li>Lien Start Date : Defaulted to Posting Date</li> <li>Lien Expiry Date : Defaulted to Deposit Maturity Date</li> <li>Lien Principal Amount : principalBalance-LienedPrincipal if((principalBalance-LienedPrincipal)&lt;lienAmount)</li> <li>Lien Interest Amount :lienAmount- Lien Principal Amount</li> <li>Principal Balance of the deposit</li> <li>Compounded Interest maintained if any</li> </ul> @param accountId @param depositId @return LienDetailsResponse
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level. @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	If Product allows Lien on <b>Principal</b> This method validates that lien amount is higher than the unliened principal balance Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" If Product is allow Lien on <b>Principal and Interest (compounded)</b> 1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest Validates the lien amount should not be less than the minimum deposit amount at product. Validate the maximum limit for principal and principal+interest defined at product validate the amount. It should not be less than zero. Throw Invalid amount entered. Please enter an amount greater than Zero. @param termDepositAccountProductAdapter @param deposit @param productId @param lienAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	This methods sets the LienPrincipal and LienInterest amount to Lien Details @param lienDetails @param updateFlag
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw Invalid amount entered. Please enter an amount greater than Zero. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowedOnPrincipal(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipal	This method return if the lien is allowed only on principal at the product level.  @param productId The product for which flag LienAllowedOnPrincipal needs to be fetched @return True if lien is allowed on principal, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowedOnPrincipalAndInterest(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level. @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	This method validates the Lien amount Entered and fills the following fields in the LienAmountDTO <ul> <li>Lien Start Date : Defaulted to Posting Date</li> <li>Lien Expiry Date : Defaulted to Deposit Maturity Date</li> <li>Lien Principal Amount : principalBalance-LienedPrincipal if((principalBalance-LienedPrincipal)<lienAmount)</li> <li>Lien Interest Amount :lienAmount- Lien Principal Amount</li> <li>Principal Balance of the deposit</li> <li>Compounded Interest maintained if any</li></ul> @param accountId @param depositId @return LienDetailsResponse
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level. @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param termDepositAccountProductAdapter @param deposit @param productId @param lienAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level.  @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	This methods sets the LienPrincipal and LienInterest amount to Lien Details @param lienDetails @param updateFlag
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level.  @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level. @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw <math>\infty</math> Invalid amount entered. Please enter an amount greater than Zero<math>\infty</math>. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level.  @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>   1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw Invalid amount entered. Please enter an amount greater than Zero. @param lienDetails

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level. @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	If Product allows Lien on <b>Principal</b>  This method validates that lien amount is higher than the unliened principal balance  Throw error "lien cannot be placed on the deposit due to lien amount being greater than unliened principal balance" <p>If Product is allow Lien on <b>Principal and Interest (compounded)</b>  1. Call the interest module for the fetch the unliened compounded interest. if lien amount is greater than the unliened principal balance + unliened compounded interest  then Throw "lien cannot be placed on the deposit due to insufficient unliened balance" <p> If lien is marked on the interest amount then intimate the Interest module to place lien on compounded interest <p> Validates the lien amount should not be less than the minimum deposit amount at product.   Validate the maximum limit for principal and principal+interest defined at product  validate the amount. It should not be less than zero.   Throw Invalid amount entered. Please enter an amount greater than Zero. @param lienDetails
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLienAllowedOnPrincipalAndInterest	This method return if the lien is allowed on both principal and Interest at the product level. @param productId The product for which flag LienAllowedOnPrincipalAndInterest needs to be fetched @return True if lien is allowed on both principal and interest, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isLienAllowedOnPrincipalAndInterest(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRedemptionSequence	This method returns the Auto Sweep Sequence for the account  1. Load the account and get the product Id.   2. Get the Auto Sweep Sequence using the productId. This will be returned using HQL.  @param productId The product for which {@link AutoRedemptionSequence} needs to be fetched @return {@link AutoRedemptionSequence} Sequence of auto redemption	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getAutoRedemptionSequence(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDExternalBankingFacilityDTO} needs to be fetched @return {@link TDExternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This method validates the external banking facilities  1.Internet Banking:- If not allowed at product level, user cannot check it at account level  2.Mobile Banking:- If not allowed at product level, user cannot check it at account level  3.Broker:- If not allowed at product level, user cannot check it at account level  4.Phone Banking / IVR:- If not allowed at product level, user cannot check it at account level  @param tdExternalBankingFacility
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDExternalBankingFacilityDTO} needs to be fetched @return {@link TDExternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This method validates the external banking facilities  1.Internet Banking:- If not allowed at product level, user cannot check it at account level  2.Mobile Banking:- If not allowed at product level, user cannot check it at account level  3.Broker:- If not allowed at product level, user cannot check it at account level  4.Phone Banking / IVR:- If not allowed at product level, user cannot check it at account level  @param tdExternalBankingFacility



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDEternalBankingFacilityDTO} needs to be fetched @return {@link TDEternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This creates a new Instance of {@link TDEternalBankingFacility} depending upon the facilities configured at offer level. @param termDepositAccount @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDEternalBankingFacilityDTO} needs to be fetched @return {@link TDEternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This method sets the default values to External Banking Facility that are defined at product, if the values are not present. @param termDepositAccount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDEternalBankingFacilityDTO} needs to be fetched @return {@link TDEternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	Populate the external banking facility, with the product default in case the caller has sent it as null. @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDExternalBankingFacilityDTO} needs to be fetched @return {@link TDExternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This method takes the domain entity as input and convert to DTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDExternalBankingFacilityDTO} needs to be fetched @return {@link TDExternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	This method takes the domain entity as input and convert to DTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getExternalBankingFacility	This method fetches the product level external banking facilities for an account  @param productId The product for which {@link TDExternalBankingFacilityDTO} needs to be fetched @return {@link TDExternalBankingFacilityDTO} Object containing flags having boolean value of external banking facilities	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getExternalBankingFacility(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionUnits	This method fetches the redemption units defined at the product level @param productId The product for which redemption units needs to be fetched @return BigDecimal redemption units	This method validates the redemption amount <ul style="list-style-type: none"><li>whether its less than principal amount</li><li>whether the balance is less than minimum deposit amount after part redemption</li><li>whether the redemption amount is in multiple of redemption units</li></ul>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionUnits	This method fetches the redemption units defined at the product level @param productId The product for which redemption units needs to be fetched @return BigDecimal redemption units	This method validates the redemption amount <ul style="list-style-type: none"><li>whether its less than principal amount</li><li>whether the balance is less than minimum deposit amount after part redemption</li><li>whether the redemption amount is in multiple of redemption units</li></ul>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionUnits	This method fetches the redemption units defined at the product level @param productId The product for which redemption units needs to be fetched @return BigDecimal redemption units	This method in case of premature partial redemption, validates the Redemption Amount with the Redemption units maintained at product level if the Redemption Amount is not in multiples of the redemption units then throws the following Exception "The Redemption Amount should be in multiples of the Redemption units maintained at the product level" @param redemptionAmount @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRedemptionUnits	This method fetches the redemption units defined at the product level @param productId The product for which redemption units needs to be fetched @return BigDecimal redemption units	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getRedemptionUnits(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPartRedemptionAllowed	This method return true if part redemption is allowed at product level @param productId The product for which flag PartRedemptionAllowed needs to be fetched @return True if part redemption is allowed, else false.	This method validates the product preference whether premature redemption is allowed or not at product level.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPartRedemptionAllowed	This method return true if part redemption is allowed at product level @param productId The product for which flag PartRedemptionAllowed needs to be fetched @return True if part redemption is allowed, else false.	This method calls validation methods depending on the type of redemption. @param payoutInstruction
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPartRedemptionAllowed	This method return true if part redemption is allowed at product level @param productId The product for which flag PartRedemptionAllowed needs to be fetched @return True if part redemption is allowed, else false.	This method returns true if Part redemption is permitted as per product definition If the instruction type is <b>Premature Redemption</b> and product does not allow the same then throw error @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPartRedemptionAllowed	This method return true if part redemption is allowed at product level @param productId The product for which flag PartRedemptionAllowed needs to be fetched @return True if part redemption is allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isPartRedemptionAllowed(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getWindowTerm	This method fetches the window term maintained at the productlevel @param productId The product for which windowTerm needs to be fetched @return Duration windowTerm	This method fetches the window Expiry term maintained at the product level and Adds it to the Deposit Date to derive the window Expiry Date for a deposit @param productId Identifier to retrieve window term @param branchId The branch for which working days are to be added @param interestStartDate The date from which the interest calculation will start. @return Window Expiry Date
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getWindowTerm	This method fetches the window term maintained at the productlevel @param productId The product for which windowTerm needs to be fetched @return Duration windowTerm	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getWindowTerm(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getWindowTerm	This method fetches the window term maintained at the product level  @param productId The product for which windowTerm needs to be fetched @return Duration windowTerm	This method fetches the lockIn term maintained at the offer level  @param offerId The offer for which lockInTerm needs to be fetched @return Duration lockInTerm
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDepositContractModificationLimit	This method return the deposit contract modification limit set at product level  @param productId The product for which deposit contract modification limit needs to be fetched @return int deposit contract modification limit	This method validates the contract modification counter with the contract modification limit stored at the product level  if limit exceeds then returns true  @param deposit @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDepositContractModificationLimit	This method return the deposit contract modification limit set at product level  @param productId The product for which deposit contract modification limit needs to be fetched @return int deposit contract modification limit	This method validates the contract modification counter with the contract modification limit stored at the product level  if limit exceeds then returns true  @param deposit @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDepositContractModificationLimit	This method return the deposit contract modification limit set at product level  @param productId The product for which deposit contract modification limit needs to be fetched @return int deposit contract modification limit	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getDepositContractModificationLimit(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getGracePeriod	This method returns grace period of term deposit  Deposit can be renewed after maturity within the grace period  @param productId The product for which renewal grace period needs to be fetched @return Integer grace period	This method retrieves pricingTicket from pricing. @param sessionContext This parameter captures the basic information about the user of the system. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies. @return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket @throws FatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getGracePeriod	This method returns grace period of term deposit  Deposit can be renewed after maturity within the grace period  @param productId The product for which renewal grace period needs to be fetched @return Integer grace period	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>then it invokes <a href="#">{@link DepositRenewalService#autoRenewDeposit}</a>. If New Deposit received from <a href="#">DepositRenewalService</a> is not Null then <a href="#">{@link DepositMaturityService#processPostMaturityRenewal}</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</p> <p>If Deposit Returned from Renewal is null then <a href="#">{@link DepositMaturityService#matureDeposit}</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured</p> <p>If PayoutResponse object is present then the method determines the type of Payout.</p> <p>If Principal Payout Instruction is Maintained then <a href="#">{@link #doPrincipalPayout}</a> is invoked.</p> <p>If PrincipalInterest Payout Instruction is Maintained then <a href="#">{@link #doPrincipalInterestPayout}</a> is invoked</p> <p>In both the cases above if Payout fails then <a href="#">{@link DepositService#matureDeposit}</a> is invoked</p> <p>Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit</p> <p><b>@param</b> sessionContext <b>@param</b> deposit <b>@throws</b> FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getGracePeriod	This method returns grace period of term deposit  Deposit can be renewed after maturity within the grace period  @param productId The product for which renewal grace period needs to be fetched @return Integer grace period	Method validates following details  <ul><li>New value date should not be less than old maturity date.</li> <li>If deposit is validated in grace period, then value date should be a date between old maturity date and current date.</li> <li>If the deposit is renewed after the grace period, the value date should be the current date</li> </ul>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getGracePeriod	This method returns grace period of term deposit  Deposit can be renewed after maturity within the grace period  @param productId The product for which renewal grace period needs to be fetched @return Integer grace period	This method logs Unclaimed mature action for the Matured Deposit. The Unclaimed Processing Date is calculated as : <p> Unclaimed Process date = {@link DepositDates#maturityDate} + {@link TDProductPreference#renewalGracePeriod} + {@link TDProductPreference#unclaimedDepTranDays}. <ol><li>If {@link UnClaimedDepositProcessingFreq#DAILY} is set at Product then the Unclaimed Processing Date is as calculated above</li> <li>If {@link UnClaimedDepositProcessingFreq#MONTHLY} then the month in which the above Date falls is determined. Then the Last Working Day of that Month is set as the Unclaimed Processing Date. Eg: If Date is 12-Jan-2011, then the Unclaimed Processing action will be logged for 31-Jan-2011</li> </p> @param applicationContext @param deposit @param calendarService @return TODO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getGracePeriod	This method returns grace period of term deposit  Deposit can be renewed after maturity within the grace period  @param productId The product for which renewal grace period needs to be fetched @return Integer grace period	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getGracePeriod(java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRepriceOnRedemptionFlag	This method returns true of repricing on redemption is allowed, else false. @param productId The product for which flag RepriceOnRedemption needs to be fetched @return True if repricing on redemption is allowed, else false.	This method do the inquiry of redemption amount. @param deposit Term deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money, com.ofss.fc.datatype.Date)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRepriceOnRedemptionFlag	This method returns true of repricing on redemption is allowed, else false. @param productId The product for which flag RepriceOnRedemption needs to be fetched @return True if repricing on redemption is allowed, else false.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getRepriceOnRedemptionFlag(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	The Business policy method validates   1. If lien allowed at product level is principal_interest, then the interest payout frequency would be at maturity.  2. If the lien allowed at product level is null, then the lien wont be allowed for the newly created deposit.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or	This method validates the lien flag.  This field can not have false in case any lien exists on the deposit.  If at the product level lienAllowed is false then this field has true then throw exception.  @param deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			none.	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	This method creates the clone of the Deposit Entity
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	This method will provide the Collateral Details for the given deposit @param sessionContext @param depositDTO @return TDCollateralDTO @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	This method will fetch the collateral liens placed on all Accounts for the given party id. External service/module will call this method to fetch the collateral values for all account and deposits by giving party ids as input. @param sessionContext @param partyIds @param collateralId @return TDCollateralResponse @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message "Account Status is closed and lien cannot be marked". <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				marked on the deposit $\cap$ $\downarrow$ . @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}. @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message $\cap$ $\downarrow$ Account Status is closed and lien cannot be marked $\cap$ $\downarrow$ . <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit $\cap$ $\downarrow$ .  @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	Checks whether lien placing is allowed at the deposit or product level.  Throw message "lien cannot be marked on the deposit" <p> Check if account is closed  if yes then throw message $\cap$ $\downarrow$ Account Status is closed and lien cannot be marked $\cap$ $\downarrow$ . <p> Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed  if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit $\cap$ $\downarrow$ .  @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}.  @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	This method generates the {@link DepositAttributeDTO} from the {@link DepositDetailInquiry} provided by {@link DepositSummaryView} from domain service {@link DepositInquiryService} @param depositDetailInquiry

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}. @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	The Business policy method validates 1. If lien allowed at product level is principal_interest, then the interest payout frequency would be at maturity. 2. If the lien allowed at product level is null, then the lien wont be allowed for the newly created deposit.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}. @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	Checks whether lien placing is allowed at the deposit or product level. Throw message "lien cannot be marked on the deposit" Check if account is closed if yes then throw message "Account Status is closed and lien cannot be marked". Check if deposit is in Matured/Unclaimed /closed status then lien placing is not allowed if yes then throw message "Deposit is in Matured/Unclaimed/closed status and lien cannot be marked on the deposit". @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}. @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	The method populates {@link Deposit} object from {@link PayinTransactionDTO} @param {@link PayinTransactionDTO} @return {@link Deposit}
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsLienAllowed	This method returns {@link LienAllowed}. @param productId The product for which {@link LienAllowed} needs to be fetched @return {@link LienAllowed} States whether lien is allowed at principal or interest or none.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsLienAllowed(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method validates the pay-out frequency for the deposit against the product.  The new pay out frequency must adhere to the minimum interest pay out frequency maintained at the product level.   Such frequency might have been modified at the deposit level, check for the frequency at product level and permit modification up-to such frequency.   <b>For e.g. </b>  At product level - Interest pay out may be quarterly modified to monthly at deposit level, which now needs to be modified to quarterly. System will permit such modification. However monthly will not be permitted.  Throw message "Modification not permitted. Interest pay out frequency at product level is breached" in case modification is not permitted as mentioned above  Interest pay out frequency can be greater than the minimum frequency maintained at Product level @param frequency {@link Frequency} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method calculates the net rate by adding the udm passed from pricing and validates the same against product level minimum and maximum  Net Interest rates eligible for a product and then updates the same over the deposit. @param arrangementNumber @param eventRateVarianceDTOArray
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	This method retrieves the InterestRuleKeyDTO from the PMU. @param productId The term deposit product id @return {@link InterestRuleKeyDTO} Object containing interest rule key parameters
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key	This method fetches payout and compounding frequency details. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing parameters required to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing deposit frequency details

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			parameters.	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling <code>fetchInterestRuleDefinition</code> from <code>InterestRuleDefinitionApplicationService</code> and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	@see <code>com.ofss.fc.app.adapter.td.ITDProductAdapter#getInterestRules(java.lang.String)</code>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling <code>fetchInterestRuleDefinition</code> from <code>InterestRuleDefinitionApplicationService</code> and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	<p> While creating product interest linkage, interest rule id is linked for each allowed currency at the product for the chosen {@link <code>com.ofss.fc.enumeration.interest.InterestDivision</code> } and {@link <code>com.ofss.fc.enumeration.interest.InterestClass</code> }.  This method will fetch IRD for the input product code and return null if not found.   This IRD will have rule attributes defined for each allowed currency at the product. This method will filter the fetched IRD for the input currency code and return only currency specific interest rule attributes. </p> @param <code>productCode</code> @param <code>currencyCode</code> @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestRules	This method fetches interest rule definition by calling fetchInterestRuleDefinition from InterestRuleDefinitionApplicationService and populates interest rule key DTO from the response. @param productId The product for which interest rule definition need to be fetched. @return Array of {@link InterestRuleKeyDTO} objects which hold interest rule key parameters.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isPMIApplicable(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getCurrencySpecificInterestRules	<p> While creating product interest linkage, interest rule id is linked for each allowed currency at the product for the chosen {@link com.ofss.fc.enumeration.interest.InterestDivision} and {@link com.ofss.fc.enumeration.interest.InterestClass}.  This method will fetch IRD for the input product code and return null if not found.  This IRD will have rule attributes defined for each allowed currency at the product. This method will filter the fetched IRD for the input currency code and return only currency specific interest rule attributes. </p> @param productCode @param currencyCode @return	This method acts as a bridge between TD module and Interest module This method is used to register the interest plan for Term Deposit with Interest module.  It also logs the payout action in TDActionLogger. @param deposit {@link DepositDTO} object containing information about deposit for which interest plan needs to be registered. @return {@link InterestRegistrationResponse} Outcome of registration.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestStartDateDecider	This method determines {@link TDInterestStartDateDecider} depending on processing date.  If processing date is true, {@link TDInterestStartDateDecider} will be depositDate. @param productId The product for which {@link TDInterestStartDateDecider} need to be fetched. @return {@link TDInterestStartDateDecider} Decides whether interestStartDate will be deposit date or value date	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInterestStartDateDecider	This method determines {@link TDInterestStartDateDecider} depending on processing date.  If processing date is true, {@link TDInterestStartDateDecider} will be depositDate. @param productId The product for which {@link TDInterestStartDateDecider} need to be fetched. @return {@link TDInterestStartDateDecider} Decides whether interestStartDate will be deposit date or value date	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getInterestStartDateDecider(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level  @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method ensures that the term provided by teller is in multiple of the term specified at product level.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method ensures that the term provided by teller is in multiple of the term specified at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	This method ensures that the term provided by teller is in multiple of the term specified at product level.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalTerm	This method returns incremental term defined at product level @param productId The product for which incremental term needs to be fetched. @return Duration incremental term	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIncrementalTerm(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	The method checks the term input by the teller.  The term is compared with the term allowed at product level for maximum value and the minimum value.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	This method validates the term of a deposit which is about to be created.  Various failure cases are :  The term is more/less than the product max/min allowed, the case fails. 
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMaximumTermLimit	This method fetches maximum days, months and years of term defined at product level @param productId The product for which maximum deposit term needs to be fetched. @return Duration maximum deposit term	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMaximumTermLimit(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method ensures that the term provided by teller is in multiple of the term specified at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	The method checks the term input by the teller.   The term is compared with the term allowed at product level for maximum value and the minimum value.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method fetches the Interest Details. This output DTO{@link TDInterestSimulationInterestRateDTO} contains {@link FactsRequiredForSimulationResponse} which stores the facts required for Simulation process, minimum deposit amount and minimum term specified for product, deposit frequency details specified at product level. @param sessionContext @param productId @param partyId @param currencyCode @return TDInterestSimulationInterestRateDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method fetches the Interest Rate for the given product from pricing module. @param sessionContext @param simulationParametersDTO @param termDepositFactsDTO @return interestRateFromPricing @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method calculates net rate applicable using base rate and userDiscretionaryMargin. @param simulationParameters {@link SimulationParameters} Holds all the input details to get rate @param pricingTicket Ticket from pricing module to get rate recommendations @return {@link InterestDetails} Object containing interest rate details
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method validates deposit term of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method ensures that the term provided by teller is in multiple of the term specified at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method validates the term of a deposit which is about to be created.  Various failure cases are :  The term is more/less than the product max/min allowed, the case fails. 
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method ensures that the term provided by teller is in multiple of the term specified at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	@param accountId @param payinAmount @param term @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method will perform the inquiry and return the {@link ConcisePayinResponse}. The depositID will not be given in the response as this will not create the deposit. @param accountId The account ID of the TD Account (Mandatory) @param payinAmount The payin amount (Mandatory) @param term The term for which the deposit is desired. If this is null, the minimum term from the product definition will be taken. (Optional) @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	This method will Create a deposit by just taking account id and amount as an input and all other parameters like term and payout frequency will be set by default the corresponding parameters set at product. This method is called by settlement module in order to do payin for TDA and NTDA modes. @param accountId holds the account id @param payinAmount {@link MoneyDTO} holds the amount for which the TD needs to funded. @return {@link PayinResponse} Containing result of payin operation.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getMinimumTermLimit	This method fetches minimum days, months and years of term defined at product level @param productId The product for which minimum deposit term needs to be fetched. @return Duration minimum deposit term	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMinimumTermLimit(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	The method checks the term input by the teller.  The term is compared with the term allowed at product level for maximum value and the minimum value.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	This method Calculates the difference between maturity date and Interest Start Date  and converts the resultant days in duration(year,Month,Day) Format depending upon the unitTermType maintained at product level @param maturityDate Maturity date of term deposit @param interestStartDate Interest start date of term deposit @param OfferId The offer id which needs for fetching UnitTermType @return Duration Deposit Term
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	This method uses interest start date and maturity date to find out the Term deposit tenor. It populates the term deposit tenor in the form of {@link DurationDTO} @param interestStartDate @param offerId @param maturityDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	This method derives term parameters based on maturity factor provided. @param termParametersDTO {@link TermParametersDTO} Object containing either maturity date or term and Maturity factor @param termStartDate Deposit start date or interest start date @param productId TD product id @return {@link TermParametersDTO} Object containing maturity date, term derived
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	This method validates the term of a deposit which is about to be created.  Various failure cases are :  The term is more/less than the product max/min allowed, the case fails. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnitTermType	This method returns {@link UnitTermType} defined at product level.  depositTerm needs to be entered depending on UnitTermType @param productId The product for which {@link UnitTermType} needs to be fetched. @return {@link UnitTermType} Specifies the unit in which depositTerm needs to be entered	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getUnitTermType(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDefaultPrincipalPayoutOption	This method returns the Default Principal Payout Option set for the Deposit at Maturity. @param productCode The product for which Default Principal Payout Option needs to be fetched. @return {@link PrincipalPayoutOption} Specifies payout option whether to redeem or reinvest or no instructions	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances &lt;/p&gt; &lt;p&gt; If Deposit Returned from Renewal is null then</p> <p>{@link DepositMaturityService#matureDeposit } is invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getDefaultPrincipalPayoutOption	This method returns the Default Principal Payout Option set for the Deposit at Maturity. @param productCode The product for which Default Principal Payout Option needs to be fetched. @return {@link PrincipalPayoutOption} Specifies payout option whether to redeem or reinvest or no instructions	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getDefaultPrincipalPayoutOption(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAddPendingInterestToPrincipal	This method inquires, whether the Add Pending Interest to Principal Flag is set for the specified product code.   It returns a Boolean value of TRUE if Pending Interest is to be added to the Principal else returns False. @param productCode The Product for which the AddPendingInterestFlag is to be checked @return Boolean	This method generates the {@link DepositBalanceDetailInquiryDTO} from the {@link Deposit} which will be set as attribute in {@link DepositAttributeDTO} @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAddPendingInterestToPrincipal	This method inquires, whether the Add Pending Interest to Principal Flag is set for the specified product code.   It returns a Boolean value of TRUE if Pending Interest is to be added to the Principal else returns False. @param productCode The Product for which the AddPendingInterestFlag is to be checked @return Boolean	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</p> <p>If Deposit Returned from Renewal is null then</p> <p>{@link #matureDeposit } is invoked, which will update the Deposit Balances and mark the Deposit as Matured</p> <p>If PayoutResponse object is present then the method determines the type of Payout.</p> <p>If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked.</p> <p>If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked</p> <p>In both the cases above if Payout fails then {@link #matureDeposit} is invoked</p> <p>Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit</p> <p>@param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAddPendingInterestToPrincipal	<p>This method inquires, whether the Add Pending Interest to Principal Flag is set for the specified product code.</p> <p>It returns a Boolean value of TRUE if Pending Interest is to be added to the Principal else returns False.</p> <p>@param productCode The Product for which the AddPendingInterestFlag is to be checked</p> <p>@return Boolean</p>	<p>@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsAddPendingInterestToPrincipal(java.lang.String)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedDepositTransferDays	This method returns unclaimed deposit transfer days defined at product level Matured deposit stays in matured status within this period if deposit has no instruction or failed instructions. After this, unclaimed deposit processing happens. @param productCode The Product for which the Unclaimed Deposit Transfer Days needs to be fetched @return Integer unclaimed deposit transfer days	This method is the Starting Point for Maturity Processing. It is Invoked from <a href="#">{@link DepositMaturityActionExecutor#executeAction}</a> . At a Broad level the method performs the following actions <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process&lt;/li&gt;</li> <li>&lt;li&gt;Invoke <a href="#">{@link DepositMaturityService#processDepositMaturity}</a>. This method returns a <a href="#">{@link MaturityResponse}</a> object. The Response object contains the following Information  <ul style="list-style-type: none"> <li>&lt;li&gt;The <a href="#">{@link RenewalResponse}</a> containing any amount which has to be renewed&lt;/li&gt;</li> <li>&lt;li&gt;The <a href="#">{@link MaturityPayoutResponse}</a> contains Principal, Interest Amount which is to be paid out and the <a href="#">{@link PayoutInstruction}</a>&lt;/li&gt;</li> <li>&lt;li&gt;The Deposit which is being Matured&lt;/li&gt;</li> </ul> </li> </ul> <li>&lt;/li&gt;The method reads the RenewalResponse. &lt;p&gt; If Not null then it invokes <a href="#">{@link DepositRenewalService#autoRenewDeposit}</a>. If New Deposit received from DepositRenewalService is not Null then <a href="#">{@link DepositMaturityService#processPostMaturityRenewal}</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances &lt;/p&gt; &lt;p&gt; If Deposit Returned from Renewal is null then <a href="#">{@link DepositMaturityService#matureDeposit}</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then <a href="#">{@link</a></li>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {<a href="#">@link #doPrincipalInterestPayout</a>} is invoked &lt;/p&gt; In both the cases above if Payout fails then {<a href="#">@link DepositService#matureDeposit</a>} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedDepositTransferDays	<p>This method returns unclaimed deposit transfer days defined at product level&lt;br&gt; Matured deposit stays in matured status within this period if deposit has no instruction or failed instructions.&lt;br&gt; After this, unclaimed deposit processing happens.&lt;br&gt; @param productCode The Product for which the Unclaimed Deposit Transfer Days needs to be fetched @return Integer unclaimed deposit transfer days</p>	<p>This method mark the deposit as mature. The Balances are updated according to the Principal Payout Option. if {<a href="#">@link PrincipalPayoutOption#NO_INSTRUCTION</a>} is set then : &lt;br&gt; &lt;ol&gt; &lt;li&gt;1. Principal Balances are moved to Uncollected Principal Balance. {<a href="#">@link Balance#uncollectedPrincipalBalance</a>} += {<a href="#">@link Balance#principleBalance</a>} + {<a href="#">@link Balance#BalancePrincipalDue</a>}&lt;/li&gt; &lt;li&gt;Interest Balances are moved to Uncollected Interest. {<a href="#">@link Balance#uncollectedInterestBalance</a>} += {<a href="#">@link Balance#BalanceInterestDue</a>}&lt;/li&gt; &lt;li&gt;The following buckets are zeroised : {<a href="#">@link Balance#principleBalance</a>}, {<a href="#">@link Balance#BalancePrincipalDue</a>}, {<a href="#">@link Balance#BalanceInterestDue</a>}&lt;/li&gt; &lt;/ol&gt; &lt;p&gt; If {<a href="#">@link PrincipalPayoutOption#RE_DEEM</a>} is set then : &lt;br&gt; &lt;li&gt;1. Principal Balances are moved to Principal Balance Due. {<a href="#">@link Balance#balancePrincipalDue</a>} += {<a href="#">@link Balance#principleBalance</a>} + {<a href="#">@link Balance#uncollectedPrincipalBalance</a>}&lt;/li&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;Interest Balances are moved to Interest Balance Due. {@link Balance#balanceInterestDue} += {@link Balance#uncollectedInterestBalance}&lt;/li&gt; &lt;li&gt;The following buckets are zeroised : {@link Balance#principleBalance}, {@link Balance#uncollectedPrincipalBalance}, {@link Balance#uncollectedInterestBalance}&lt;/li&gt; &lt;/ol&gt; &lt;/p&gt;           @param deposit</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedDepositTransferDays	<p>This method returns unclaimed deposit transfer days defined at product level&lt;br&gt; Matured deposit stays in matured status within this period if deposit has no instruction or failed instructions.&lt;br&gt; After this, unclaimed deposit processing happens.&lt;br&gt; @param productCode The Product for which the Unclaimed Deposit Transfer Days needs to be fetched @return Integer unclaimed deposit transfer days</p>	<p>This method logs Unclaimed mature action for the Matured Deposit. The Unclaimed Processing Date is calculated as : &lt;p&gt; Unclaimed Process date = {@link DepositDates#maturityDate} + {@link TDProductPreference#renewalGracePeriod} + {@link TDProductPreference#unclaimedDepTranDays}. &lt;ol&gt; &lt;li&gt;If {@link UnClaimedDepositProcessingFreq#DAILY} is set at Product then the Unclaimed Processing Date is as calculated above&lt;/li&gt; &lt;li&gt;If {@link UnClaimedDepositProcessingFreq#MONTHLY} then the month in which the above Date falls is determined. Then the Last Working Day of that Month is set as the Unclaimed Processing Date. Eg: If Date is 12-Jan-2011, then the Unclaimed Processing action will be logged for 31-Jan-2011&lt;/li&gt; &lt;/ol&gt; @param applicationContext @param deposit @param calendarService @return TODO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedDepositTransferDays	This method returns unclaimed deposit transfer days defined at product level  Matured deposit stays in matured status within this period if deposit has no instruction or failed instructions.  After this, unclaimed deposit processing happens.  @param productCode The Product for which the Unclaimed Deposit Transfer Days needs to be fetched @return Integer unclaimed deposit transfer days	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getUnclaimedDepositTransferDays(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedProcessFrequency	This method returns {@link UnClaimedDepositProcessingFreq} defined at product level  @param productCode The Product for which the Unclaimed Process Frequency needs to be calculated @return {@link UnClaimedDepositProcessingFreq} specifies unclaimed deposit processing frequency	Based on the flag isRedemptionRenewalforUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePricipalDue and UncollectedPricipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedProcessFrequency	<p>This method returns <a href="#">UnClaimedDepositProcessingFreq</a> defined at product level</p> <p>@param productCode The Product for which the Unclaimed Process Frequency needs to be calculated</p> <p>@return <a href="#">UnClaimedDepositProcessingFreq</a> specifies unclaimed deposit processing frequency</p>	<p>This method is the Starting Point for Maturity Processing. It is Invoked from <a href="#">DepositMaturityActionExecutor#executeAction</a>. At a Broad level the method performs the following actions</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>&lt;li&gt;Invoke <a href="#">DepositMaturityService#processDepositMaturity</a>.</li> </ul> <p>This method returns a <a href="#">MaturityResponse</a> object. The Response object contains the following Information</p> <ul style="list-style-type: none"> <li>&lt;li&gt;The <a href="#">RenewalResponse</a> containing any amount which has to be renewed</li> <li>&lt;li&gt;The <a href="#">MaturityPayoutResponse</a> contains Principal, Interest Amount which is to be paid out and the <a href="#">PayoutInstruction</a></li> <li>&lt;li&gt;The Deposit which is being Matured</li> </ul> <p>The method reads the <a href="#">RenewalResponse</a>. If Not null then it invokes <a href="#">DepositRenewalService#autoRenewDeposit</a>. If New Deposit received from <a href="#">DepositRenewalService</a> is not Null then <a href="#">DepositMaturityService#processPostMaturityRenewal</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</p> <p>If Deposit Returned from Renewal is null then <a href="#">DepositMaturityService#matureDeposit</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured</p> <p>If PayoutResponse object is present then the method determines the type of Payout. If Principal Payout Instruction is Maintained then <a href="#">MaturityResponse</a></p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedProcessFrequency	<p>This method returns {@link UnClaimedDepositProcessingFreq} defined at product level&lt;br&gt; @param productCode The Product for which the Unclaimed Process Frequency needs to be calculated @return {@link UnClaimedDepositProcessingFreq} specifies unclaimed deposit processing frequency</p>	<p>This method mark the deposit as mature. The Balances are updated according to the Principal Payout Option. if {@link PrincipalPayoutOption#NO_INSTRUCTION} is set then : &lt;br&gt; &lt;ol&gt; &lt;li&gt;1. Principal Balances are moved to Uncollected Principal Balance. {@link Balance#uncollectedPrincipalBalance} += {@link Balance#principleBalance} + {@link Balance#BalancePrincipalDue}&lt;/li&gt; &lt;li&gt;Interest Balances are moved to Uncollected Interest. {@link Balance#uncollectedInterestBalance} += {@link Balance#BalanceInterestDue}&lt;/li&gt; &lt;li&gt;The following buckets are zeroised : {@link Balance#principleBalance}, {@link Balance#BalancePrincipalDue}, {@link Balance#BalanceInterestDue}&lt;/li&gt; &lt;/ol&gt; &lt;p&gt; If {@link PrincipalPayoutOption#RE_DEEM} is set then : &lt;br&gt; &lt;li&gt;1. Principal Balances are moved to Principal Balance Due. {@link Balance#balancePrincipalDue} += {@link Balance#principleBalance} + {@link Balance#uncollectedPrincipalBalance}&lt;/li&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;Interest Balances are moved to Interest Balance Due. {@link Balance#balanceInterestDue} += {@link Balance#uncollectedInterestBalance}&lt;/li&gt; &lt;li&gt;The following buckets are zeroised : {@link Balance#principleBalance}, {@link Balance#uncollectedPrincipalBalance}, {@link Balance#uncollectedInterestBalance}&lt;/li&gt; &lt;/ol&gt; &lt;/p&gt;           @param deposit</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedProcessFrequency	<p>This method returns {@link UnClaimedDepositProcessingFreq} defined at product level&lt;br&gt;           @param productCode The Product for which the Unclaimed Process Frequency needs to be calculated            @return {@link UnClaimedDepositProcessingFreq} specifies unclaimed deposit processing frequency</p>	<p>This method logs Unclaimed mature action for the Matured Deposit. The Unclaimed Processing Date is calculated as : &lt;p&gt; Unclaimed Process date = {@link DepositDates#maturityDate} + {@link TDProductPreference#renewalGracePeriod} + {@link TDProductPreference#unclaimedDepTranDays}. &lt;ol&gt; &lt;li&gt;If {@link UnClaimedDepositProcessingFreq#DAILY} is set at Product then the Unclaimed Processing Date is as calculated above&lt;/li&gt; &lt;li&gt;If {@link UnClaimedDepositProcessingFreq#MONTHLY} then the month in which the above Date falls is determined. Then the Last Working Day of that Month is set as the Unclaimed Processing Date. Eg: If Date is 12-Jan-2011, then the Unclaimed Processing action will be logged for 31-Jan-2011&lt;/li&gt; &lt;/p&gt;           @param applicationContext @param deposit @param calendarService @return TODO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getUnclaimedProcessFrequency	This method returns {@link UnClaimedDepositProcessingFreq} defined at product level  @param productCode The Product for which the Unclaimed Process Frequency needs to be calculated @return {@link UnClaimedDepositProcessingFreq} specifies unclaimed deposit processing frequency	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getUnclaimedProcessFrequency(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDefaultInterestPayoutOption	This method returns {@link IntPayoutOptionType} defined at product level  @param productId The Product for which {@link IntPayoutOptionType} needs to be fetched @return {@link IntPayoutOptionType} specifies payout option whether to redeem or reinvest or no instructions	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDefaultInterestPayoutOption	This method returns {@link IntPayoutOptionType} defined at product level  @param productId The Product for which {@link IntPayoutOptionType} needs to be fetched @return {@link IntPayoutOptionType} specifies payout option whether to redeem or reinvest or no instructions	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getDefaultInterestPayoutOption(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDiscretionaryMarginApplicableOnRenewal	This method returns true if discretionary margin is applicable on renewal of term deposit, else false  @param productId The Product for which the flag DiscretionaryMarginApplicableOnRenewal needs to be fetched @return Boolean	This method retrieves pricingTicket from pricing. @param sessionContext This parameter captures the basic information about the user of the system. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies. @return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket @throws FatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getDiscretionaryMarginApplicableOnRenewal	This method returns true if discretionary margin is applicable on renewal of term deposit, else false  @param productId The Product for which the flag DiscretionaryMarginApplicableOnRenewal needs to be fetched @return Boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getDiscretionaryMarginApplicableOnRenewal(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRateFromMaturity	This method returns {@link RenewalRateFromMaturityDate} defined at product level @param productId The Product for which {@link RenewalRateFromMaturityDate} needs to be fetched @return {@link RenewalRateFromMaturityDate} Specifies renewal rate is current, higher, lower or none from maturity date.	This method retrieves pricingTicket from pricing. @param sessionContext This parameter captures the basic information about the user of the system. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} holds field data relevant to inquiry before actual deposit open. It contains fields like interest start date, principal balance, deposit date, account Id, deposit id, maturity date, frequencies. @return RateOffersFromPricing {@link RateOffersFromPricing} contains pricing ticket @throws FatalException {@link FatalException} holds the exception coming from the transaction, if any @see com.ofss.fc.app.td.service.deposit.IDepositRenewalApplicationService#getOffersFromPricing(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRateFromMaturity	This method returns {@link RenewalRateFromMaturityDate} defined at product level @param productId The Product for which {@link RenewalRateFromMaturityDate} needs to be fetched @return {@link RenewalRateFromMaturityDate} Specifies renewal rate is current, higher, lower or none from maturity date.	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getRenewalRateFromMaturity(java.lang.String)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	Based on the flag isRedemptionRenewalForUnclaimedDepositAllowed the method processUnclaimedDeposit will process the balancePrincipalDue and UncollectedPrincipalBalance to UnclaimedPrincipal and call DepositRepository to save the deposit object @param deposit {@link Deposit} Unclaimed deposit object to be processed @param applicationContext Application Context which will be set in batch process. @param dueDate
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	This method validates the deposit status of TD
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	For an account to be renewed it must fulfill following conditions : 1. Its deposit status should be matured/Unclaimed. 2. RenewalRedemptionFlag is a flag at product level. The value of the flag decides if deposit in unclaimed state can be renewed. 3. If it not allowed by the RenewalRedemptionFlag, then a deposit in Matured state can be renewed.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	This method validates the deposit status whether it is unclaimed and redemptionRenewal for this is false

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false  @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	For an account to be renewed it must fulfill following conditions :   Its deposit status should be matured/Unclaimed.   1. RenewalRedemptionFlag is a flag at product level. The value of the flag decides if deposit in unclaimed state can be renewed.   2. If it not allowed by the RenewalRedemptionFlag, then a deposit in Matured state can be renewed.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false  @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	This method validates the deposit status.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getRenewalRedemptionFlag	This method returns true if renewal on redemption is allowed for unclaimed account, else false  @param productId The Product for which the flag AllowRedemptionRenewalForUnclaimedAccount needs to be fetched @return Boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getRenewalRedemptionFlag(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getAccountRetentionDays	This method gives the account retention days.  @param productId The Product for which the account retention days needs to be fetched @return int account retention days	@param termDepositAccountOpenDTO @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAccountRetentionDays	This method gives the account retention days.  @param productId The Product for which the account retention days needs to be fetched @return int account retention days	When the account was created, an action was logged for it to be deleted if no deposit is opened in it in the retention days span. This method removes that logged action for the account. @param payinTransactionDTO {@link PayinTransactionDTO} containing details of payin transaction
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAccountRetentionDays	This method gives the account retention days.  @param productId The Product for which the account retention days needs to be fetched @return int account retention days	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getAccountRetentionDays(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPMIAplicable	This method returns true, if PMI is applicable.  @param productId The Product for which the flag PMIAplicable needs to be fetched @return boolean	This method performs a redemption. @param depositTerm deposit details @param redemptionTransactionDTO Details of redemption transaction @return {@link RedemptionResponseDetailsDTO} Object containing details of redemption after actual redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#doRedemption()
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPMIAplicable	This method returns true, if PMI is applicable.  @param productId The Product for which the flag PMIAplicable needs to be fetched @return boolean	This method do the inquiry of redemption amount. @param depositTerm deposit details @param redemptionDetailsInquiryRequest {@link RedemptionRequestDTO} Object containing information required for inquiry of redemption details @return RedemptionDetailsDTO {@link RedemptionDetailsDTO} Object containing details of redemption after inquiry of redemption @see com.ofss.fc.domain.td.strategy.redemption.RedemptionStrategy#inquireRedemptionAmount(com.ofss.fc.domain.td.entity.deposit.Deposit, com.ofss.fc.datatype.Money, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPMIApplicable	This method returns true, if PMI is applicable.  @param productId The Product for which the flag PMIApplicable needs to be fetched @return boolean	This method creates and returns an array of TDInterestDivisionDTO. On Deposit Maturity Interest has to be informed that the Deposit has matured and that Regular Interest is no longer to be applied. This is done by invoking a performInterestComputation method by passing a Zero Balance Base Amount and Interest Rate Zero for Regular Interest Division. Also If the Deposit is eligible for PMI Interest, then performInterestComputation on Interest is triggered for Deposit Uncollected Principal with Interest Division Post Maturity @param maturityProcessInformation @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isPMIApplicable	This method returns true, if PMI is applicable.  @param productId The Product for which the flag PMIApplicable needs to be fetched @return boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#isPMIApplicable(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestProdCode	The Product to which the Deposit is to be Reinvested @param productId The Product for which the reinvestProdCode needs to be fetched @return the reinvestProdCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestProdCode(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestProdCode	The Product to which the Deposit is to be Reinvested @param productId The Product for which the reinvestProdCode needs to be fetched @return the reinvestProdCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestProdCode(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method validates the business policies 1. Call validateParties() { <a href="#">@link #validateParties(PartyAccountRelationshipDetails[])</a> } 2. Call validateOpenDate() { <a href="#">@link #validateOpenDate(Date)</a> } 3. Call validateMinorAccountStatus() { <a href="#">@link #validateMinorAccountStatus(TermDepositAccount)</a> } 4. Call validateProduct() { <a href="#">@link #validateProduct(String)</a> } 5. Call validateBranchCode() { <a href="#">@link #validateBranchCode(String)</a> } 6. Call validateCurrency() { <a href="#">@link #validateCurrency(String)</a> } @param termDepositAccount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method will validate the product code if the product code is active and not expired @param productId
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method holds validateBusinessPolicy method that does the needed validations for blocking an account/deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method validates the product preference whether premature redemption is allowed or not at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method holds validateBusinessPolicy method that does the needed validations for blocking an account/deposit

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method validates the product preference whether premature redemption is allowed or not at product level.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product   Refactored on March 27 to change the input from productGroupLinkageDTO to ContractDetailComputationRequest. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, ContractDetailComputationRequest, contractDetailComputationRequest, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This service will validate the term deposit product group details with the Term Deposit module. Various parameters of the term deposit such as the requested amount, the requested term, interest payout frequency and the term multiplicity i.e. the term provided is in the multiples of the term specified at the product level are validated against the restrictions that are setup on the product @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, TermDepositProductGroupLinkageDTO, productGroupLinkageDTO, , Indicates the product level attributes such as requested term, requested amount, currency and the payout frequency specified by the applicant(s) @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	This method will validate the product code.if the product code is active and not expired  @param productId
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateProduct	This method validates product code @param productCode The code which needs to be validated @return boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#validateProduct(java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getisAllowReinvestThresholdValue	Flag to determine whether to allow Reinvest if renewal amount breaches maximum or minimum limit @param productId The Product for which the flag AllowReinvestThresholdValue needs to be fetched @return boolean	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances </p> <p> If Deposit Returned from Renewal is null then {@link DepositMaturityService#matureDeposit} is invoked, which will update the Deposit Balances and mark the Deposit as Matured </p> </li> <li>If PayoutResponse object is present then the method determines the type of Payout. <p> If Principal Payout Instruction is Maintained then {@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getisAllowReinvestThresholdValue	<p>Flag to determine whether to allow Reinvest if renewal amount breaches maximum or minimum limit @param productId The Product for which the flag AllowReinvestThresholdValue needs to be fetched @return boolean</p>	<p>This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getisAllowReinvestThresholdValue	<p>Flag to determine whether to allow Reinvest if renewal amount breaches maximum or minimum limit @param productId The Product for which the flag AllowReinvestThresholdValue needs to be fetched @return boolean</p>	<p>@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getisAllowReinvestThresholdValue(java.lang.String)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method creates the clone of the Deposit Entity
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method generates the {@link DepositAttributeDTO} from the {@link DepositDetailInquiry} provided by {@link DepositSummaryView} from domain service {@link DepositInquiryService} @param depositDetailInquiry
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol style="list-style-type: none"> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol style="list-style-type: none"> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> </ol> <p>The method reads the RenewalResponse. &lt;p&gt; If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances</p> <p>If Deposit Returned from Renewal is null then</p> <p>{@link DepositMaturityService#matureDeposit } is invoked, which will update the Deposit Balances and mark the Deposit as Matured</p> <p>If PayoutResponse object is present then the method determines the type of Payout.</p> <p>If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked.</p> <p>If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked</p> <p>In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked</p> <p>Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit</p> <p>@param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method assembles the auto renewal and Force Renewal terms from deposit only if set as zero at product. @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method creates a New Deposit in the Account as specified in @param accountResponse.accountId. The new Deposit parameters are read from existing Deposit which is received as a parameter. @param applicationContext @param accountResponse @param deposit @param renewalAmount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	This method creates a New Deposit in the Account as specified in @param accountResponse.accountId. The new Deposit parameters are read from existing Deposit which is received as a parameter. @param applicationContext @param accountResponse @param deposit @param renewalAmount @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	The method populates {@link Deposit} object from {@link PayinTransactionDTO} @param {@link PayinTransactionDTO} @return {@link Deposit}
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAutoRenewalTerm	Term of Reinvested Deposit @param productId The Product for which auto renewal term needs to be fetched @return Duration autoRenewalTerm	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getAutoRenewalTerm(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method creates the clone of the Deposit Entity

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method generates the {@link DepositAttributeDTO} from the {@link DepositDetailInquiry} provided by {@link DepositSummaryView} from domain service {@link DepositInquiryService} @param depositDetailInquiry
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol style="list-style-type: none"> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol style="list-style-type: none"> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> </ol> <p>The method reads the RenewalResponse.   If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances   If Deposit Returned from Renewal is null then  {@link DepositMaturityService#matureDeposit} is</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				invoked, which will update the Deposit Balances and mark the Deposit as Matured </p> </li> <li>If PayoutResponse object is present then the method determines the type of Payout. <p> If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked. </p> <p> If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked </p> In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked</li> </ol> Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method assembles the auto renewal and Force Renewal terms from deposit only if set as zero at product. @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	This method creates a New Deposit in the Account as specified in @param accountResponse.accountId. The new Deposit parameters are read from existing Deposit which is received as a parameter. @param applicationContext @param accountResponse @param deposit @param renewalAmount @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	The method populates {@link Deposit} object from {@link PayinTransactionDTO} @param {@link PayinTransactionDTO} @return {@link Deposit}
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getForceRenewalTerm	Term Of new Deposit if Deposit is Force Renewed @param productId The Product for which force renewal term needs to be fetched @return Duration forceRenewalTerm	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getForceRenewalTerm(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	This method ensures that the principalBalance is in range of amount defines at product level
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	This method validates initial investment amount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	This method validates initial investment amount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	If the new deposit amount does not satisfy the minimum amount for the TD as set at the Product level  Throw message as " Modification of the contract not permitted. Deposit does not satisfy the minimum deposit amount" <p> If the new deposit amount does not satisfy the maximum amount for the TD as set at the Product level   Throw message as ⌈⌋ Modification of the contract not permitted. Deposit does not satisfy the maximum deposit amount criteria⌈⌋ @param newPrincipalAmount {@link Balance#getPrincipleBalance()}



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIncrementalAmount	The amount specified at product level.  The payin amount needs to be the incremental of this term. @param productId The Product for which incremental amount needs to be fetched @param currencyCode The currency allowed for the given product @return BigDecimal incremental amount	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIncrementalAmount(java.lang.String, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchTDIPP Frequencies	The method returns the payout frequencies. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing details to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing details of interest compounding frequency, allowed frequencies and default payout frequency	This method validates the pay-out frequency for the deposit against the product.  The new pay out frequency must adhere to the minimum interest pay out frequency maintained at the product level.   Such frequency might have been modified at the deposit level, check for the frequency at product level and permit modification up-to such frequency.   <b>For e.g. </b>  At product level - Interest pay out may be quarterly modified to monthly at deposit level, which now needs to be modified to quarterly. System will permit such modification. However monthly will not be permitted.  Throw message "Modification not permitted. Interest pay out frequency at product level is breached" in case modification is not permitted as mentioned above  Interest pay out frequency can be greater than the minimum frequency maintained at Product level @param frequency {@link Frequency} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchTDIPP Frequencies	The method returns the payout frequencies. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing details to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing details of interest compounding frequency, allowed frequencies and default payout frequency	This method fetches payout and compounding frequency details. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing parameters required to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing deposit frequency details
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchTDIPP Frequencies	The method returns the payout frequencies. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing details to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing details of interest compounding frequency, allowed frequencies and default payout frequency	This method will Create a deposit by just taking account id and amount as an input and all other parameters like term and payout frequency will be set by default the corresponding parameters set at product. This method is called by settlement module in order to do payin for TDA and NTDA modes. @param accountId holds the account id @param payinAmount {@link MoneyDTO} holds the amount for which the TD needs to funded. @return {@link PayinResponse} Containing result of payin operation.
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchTDIPP Frequencies	The method returns the payout frequencies. @param contractDetailComputationRequest {@link ContractDetailComputationRequest} Object containing details to fetch frequency details @return {@link DepositFrequencyDetailsDTO} Object containing details of interest compounding frequency, allowed frequencies and default payout	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#fetchTDIPP Frequencies(com.ofss.fc.app.td.dto.transaction.payin.ContractDetailComputationRequest)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			frequency	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	inquireProduct	This method is used to inquire the savingspreferencesApplicationService :: fetchSavingspreferences so as to populate all the details required to set different preferences at the time of account opening @param productId Term deposit product id @return {@link ProductDetailsDTO} Object containing id and name of the product @throws FatalException	This method will fetch all the product details for Term Deposit product class @param offerId @return Array of ProductDetailsForInterestSimulationDTO @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	inquireProduct	This method is used to inquire the savingspreferencesApplicationService :: fetchSavingspreferences so as to populate all the details required to set different preferences at the time of account opening @param productId Term deposit product id @return {@link ProductDetailsDTO} Object containing id and name of the product @throws FatalException	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	inquireProduct	This method is used to inquire the savingspreferencesApplicationService :: fetchSavingspreferences so as to populate all the details required to set different preferences at the time of account opening @param productId Term deposit product id @return {@link ProductDetailsDTO} Object containing id and name of the product @throws FatalException	The method fetches the following parameters Branch Name, Product Name ,Party Name @param termDepositAccountDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	inquireProduct	This method is used to inquire the savingspreferencesApplicationService :: fetchSavingspreferences so as to populate all the details required to set different preferences at the time of account opening @param productId Term deposit product id @return {@link ProductDetailsDTO} Object containing id and name of the product @throws FatalException	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#inquireProduct(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method will create a pricing plan within pricing module, by calling a function to link  the arrangement number to the recommendation created earlier on the screen.  The recommendation is stored in pricing and is referenced by the ticketNumber which is an input parameter  The implementation will handle the case where the ticket number is null by itself   calling for a recommendation from pricing and thus getting a ticket number. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing facts required to create pricing plan @param ticketNumber The pricing ticket required for pricing plan @param effectiveDate The date which is either deposit or interest start date @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#createPricingPlan(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method calls pricing to retrieve admin fees and pricing details<b> @param depositFactsDTO {@link TermDepositFactsDTO} object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The deposit or interest start date @param ticketNumber The pricing ticket number @return {@link PriceOfferingResponse} Object containing pricing ticket @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getPriceOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	Methods to call for an offering from pricing. This method takes a contract context from the calling module. Begins a dialogue which means get a pricing ticket from pricing. Calls for the interest rate offerings from pricing. and then returns the pricing ticket to the caller if all has gone smoothly. If not then a business exception is thrown The aim of this method if to begin a dialogue with pricing. the dialogue is closed by calling LinkVarianceandEndDialogue with the pricing ticket instantiated here. @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate The date which is either deposit or interest start date @param ticketNumber The pricing ticket number @return {@link RateOfferingResponse} Object containing pricing ticket number @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getRateOffering(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method will be called before account or deposit is opened   for adding rate variances to the pricing recommendation @param depositFactsDTO {@link TermDepositFactsDTO} Object containing TD parameters like partyId, productId, FactDetails @param effectiveDate Interest start date @param ticketNumber The pricing ticket number @param eventRateVarianceDTOs Array of {@link EventRateVarianceDTO} objects containing details of rate variances @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#mergeRateVarianceToRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String, com.ofss.fc.app.pricing.dto.contract.EventRateVarianceDTO[])
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method will fetch rates from pricing for Pricing recommendation created for deposit @param termDepositFactsDTO {@link TermDepositFactsDTO} Containing details of facts required for fetching net rates @param evaluationDate Date for which rate needs to be fetched @param pricingTicket Pricing ticket required for interaction with pricing module @return {@link InterestDetailsDTO} Containing information about interest rates @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#getNetRateAtRecommendation(com.ofss.fc.app.adapter.td.o.TermDepositFactsDTO, com.ofss.fc.datatype.Date, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	Re-prices the complete {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing offerings, re-evaluating at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. <p> The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. </p> @param sessionContext The session context @param depositFactsDTO {@link TermDepositFactsDTO} Object containing information about deposit for which pricing plan is to be created registered. @param arrangementNumber The arrangement number

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param effectiveDate The date on which to reprice @see com.ofss.fc.app.adapter.td.ITDPricingAdapter#repriceAll(com.ofss.fc.app.adapter.td.dto.TermDepositFactsDTO, java.lang.String, com.ofss.fc.datatype.Date)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method uses IDDAProductAdapter and fetches productGroupId for the product. @param productId @return productGroupId
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price-offerings, and are NOT reinstated with the new price offerings. @param sessionContext @param termDepositFactsDTO @param repriceDate
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns the DepositWorkListDTO @param arrangementNumber @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method will retrieve the facts that are required for simulation  Facts are required to obtain TD interest rates defined in Pricing module.  Product ids are input for this function as Product level and Bank level price policy charts will be searched for list of facts<> @param simulationFactInquiryRequest {@link SimulationFactInquiryRequest} Object containing parameters required for fetching facts @return Set of {@link FactDetails} objects containing details of facts required for simulation @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method calls the OutboundDocumentApplicationService.generateSetOfDocument for printing. @param sessionContext @param certificatePrintFactsDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method is used to fetch all the accounts for given party id and it also fetches product group code for the product id of account. @param partyId Term deposit party id @return Array of {@link TermDepositAccount} objects fetched
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductGroupId	This method returns product group id for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getProductGroupId(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateCurrencyDetails	This method will validate whether currency is allowed at product level or not.  @param productId The Product for which currency needs to be validated @param currencyCode The currency which needs to be validated @return boolean	This method validates the business policies  1. Call validateParties() {@link #validateParties(PartyAccountRelationshipDetails[])}  2. Call validateOpenDate() {@link #validateOpenDate(Date)}  3. Call validateMinorAccountStatus() {@link #validateMinorAccountStatus(TermDepositAccount)}  4. Call validateProduct() {@link #validateProduct(String)}  5. Call validateBranchCode() {@link #validateBranchCode(String)}  6. Call validateCurrency() {@link #validateCurrency(String)}  @param termDepositAccount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateCurrencyDetails	This method will validate whether currency is allowed at product level or not.  @param productId The Product for which currency needs to be validated @param currencyCode The currency which needs to be validated @return boolean	This method will validate whether currency is allowed at product level or not. @param productId @param currencyCode
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	validateCurrencyDetails	This method will validate whether currency is allowed at product level or not.  @param productId The Product for which currency needs to be validated @param currencyCode The currency which needs to be validated @return boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#validateCurrencyDetails(java.lang.String, java.lang.String)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLeadDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	This method logs the actions when the corresponding parameters like balance,term,PayoutFrequency is modified @param existDeposit {@link Deposit} Term deposit details before contract modification @param deposit {@link Deposit} Term deposit details after contract modification @param transactionReferenceNumber Transaction reference number @param transactionSubsequenceNumber Transaction sequence number
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLeadDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	This method raises an alert for the deposit being matured and also logs an action for maturity lag alert and   unclaimed lead alert @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLeadDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLeadDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getAlertLeadDays(java.lang.String, TDEventName)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLeadDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getAlertLagDays(java.lang.String, TDEventName)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLagDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	This method raises an alert for the deposit being moved to unclaimed status and also logs an action for unclaimed lag alert @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getAlertLagDays	This method fetches the Alert Lead days @param productId The Product for which alert lead days needs to be fetched @param eventName {@link TDEventName} @return int alert lead days	This method raises an alert for the deposit being matured and also logs an action for maturity lag alert and unclaimed lead alert @param deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLedgerCode	This method fetches the Ledger Code maintained for a Product for the given Account Role @param productId @param accountRole @return	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLedgerCode	This method fetches the Ledger Code maintained for a Product for the given Account Role @param productId @param accountRole @return	This method is used to payin deposit.  Method will be called from screen. @param payinTransactionDTO {@link PayinTransactionDTO} contains details of the fields need to open a deposit. @param raiseAccountingFlag Flag which indicates whether accounting entries needs to be passed or not. @return {@link PayinResponse} object containing outcome of Payin Transaction
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLedgerCode	This method fetches the Ledger Code maintained for a Product for the given Account Role @param productId @param accountRole @return	This method fetches the Ledger Code maintained for a Product for the given Account Role @param productId @param accountRole @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLedgerCode	This method fetches the Ledger Code maintained for a Product for the given Account Role @param productId @param accountRole @return	This method is used to fetch Clearing Bridge Ledger @return Clearing Bridge Ledger @see com.ofss.fc.app.adapter.td.ITDProductAdapter#fetchClearingBridgeLedger()

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getHasBroker	This method returns whether Broker is enabled for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method validates the external banking facilities 1.Internet Banking:- If not allowed at product level, user cannot check it at account level 2.Mobile Banking:- If not allowed at product level, user cannot check it at account level 3.Broker:- If not allowed at product level, user cannot check it at account level 4.Phone Banking / IVR:- If not allowed at product level, user cannot check it at account level @param TDExternalBankingFacility
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getHasBroker	This method returns whether Broker is enabled for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This method validates the external banking facilities 1.Internet Banking:- If not allowed at product level, user cannot check it at account level 2.Mobile Banking:- If not allowed at product level, user cannot check it at account level 3.Broker:- If not allowed at product level, user cannot check it at account level 4.Phone Banking / IVR:- If not allowed at product level, user cannot check it at account level @param TDExternalBankingFacility
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getHasBroker	This method returns whether Broker is enabled for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	This creates a new Instance of {@link TDExternalBankingFacility} depending upon the facilities configured at offer level. @param termDepositAccount @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getHasBroker	This method returns whether Broker is enabled for the given product @param productId The Product for which product group id needs to be fetched @return String product group id	Populate the external banking facility, with the product default in case the caller has sent it as null. @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getHasBroker	This method returns whether Broker is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns whether Broker is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInternetBankingAllowed	This method returns whetherInternet Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	Populate the external banking facility, with the product default in case the caller has sent it as null. @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getInternetBankingAllowed	This method returns whetherInternet Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns whetherInternet Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMobileBankingAllowed	This method returns whether Mobile Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	Populate the external banking facility, with the product default in case the caller has sent it as null. @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getMobileBankingAllowed	This method returns whether Mobile Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns whether Mobile Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPhoneBankingAllowed	This method returns whether Phone Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	Populate the external banking facility, with the product default in case the caller has sent it as null. @param termDepositAccountOpenDTO @throws FatalException @since OBP2.2 JIRA OBP22IUT-10548
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPhoneBankingAllowed	This method returns whether Phone Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns whether Phone Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getPhoneBankingAllowed	This method returns whether Phone Banking is enabled for the given product  @param productId The Product for which product group id needs to be fetched @return String product group id	This method returns currency attributes for the given product  @param productId The Product for which product group id needs to be fetched @return String product id
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchClearingBridgeLedger	This method is used to fetch Clearing Bridge Ledger @return Clearing Bridge Ledger	This is meant to change the depositStatus of a deposit to regular from fundAwaited, If the isSuccess comes true else close the deposit. @param accountId {@link String} holds the account Id under which the deposit is present. @param depositId {@link Integer} holds the deposit id of the deposit. @param isSuccess {@link Boolean} the flag from settlement, which will tell us to change the deposit status. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchClearingBridgeLedger	This method is used to fetch Clearing Bridge Ledger @return Clearing Bridge Ledger	This method is used to fetch Clearing Bridge Ledger @return Clearing Bridge Ledger @see com.ofss.fc.app.adapter.td.ITDProductAdapter#fetchClearingBridgeLedger()
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchAllTDProducts	This method is used to fetch Clearing Bridge Ledger @return Clearing Bridge Ledger	This method is used to fetch all TD products @return Product Details @see com.ofss.fc.app.adapter.td.ITDProductAdapter#fetchAllTDProducts()

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnlytAtMaturity	This method returns true if waive penalty at maturity allowed else false. @param productId Term Deposit product id @return boolean	<p>This method is the Starting Point for Maturity Processing. It is Invoked from <a href="#">{@link DepositMaturityActionExecutor#executeAction}</a>. At a Broad level the method performs the following actions</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process&lt;/li&gt; &lt;li&gt;Invoke <a href="#">{@link DepositMaturityService#processDepositMaturity}</a>. This method returns a <a href="#">{@link MaturityResponse}</a> object. The Response object contains the following Information &lt;ol&gt; &lt;li&gt;The <a href="#">{@link RenewalResponse}</a> containing any amount which has to be renewed&lt;/li&gt; &lt;li&gt;The <a href="#">{@link MaturityPayoutResponse}</a> contains Principal, Interest Amount which is to be paid out and the <a href="#">{@link PayoutInstruction}</a>&lt;/li&gt; &lt;li&gt;The Deposit which is being Matured&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;The method reads the RenewalResponse. &lt;p&gt; If Not null then it invokes <a href="#">{@link DepositRenewalService#autoRenewDeposit}</a>. If New Deposit received from DepositRenewalService is not Null then <a href="#">{@link DepositMaturityService#processPostMaturityRenewal}</a> is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances &lt;/p&gt; &lt;p&gt; If Deposit Returned from Renewal is null then <a href="#">{@link DepositMaturityService#matureDeposit}</a> is invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then <a href="#">{@link</a></li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>#doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsWaivePnltyAtMaturity	This method returns true if waive penalty at maturity allowed else false. @param productId Term Deposit product id @return boolean	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsWaivePnltyAtMaturity(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getContractModificationInterestRecomputeFactor	This method fetches the deposit contract modification recompute factor defined at product level, It could be one of the following   <ul> <li>New Rate for Lapsed Period_New Rate for Remaining Period(NN) :   1.For Lapse period : New Interest rate is fetched from pricing based on lapse no. of days.  Interest accrued till date is reversed and new interest is computed based on new rate.  2. For Remanig term : New interest rate will be fetched from pricing based on applicable facts.  New interest rate will be applicable for remaining term of TD</li>	This method fetches the Interest Recompute factor defined at product @param offerId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<ul style="list-style-type: none"> <li>&lt;li&gt;Existing Rate for Lapsed Period_New Rate for Remaining Period(EE) :&lt;br&gt; 1.For Lapsed Period: Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation).&lt;br&gt; 2.For Remaining Term:New interest rate will be fetched from pricing based on applicable facts.&lt;br&gt; New interest rate will be applicable for remaining term of TD.&lt;/li&gt;</li> <li>&lt;li&gt;Existing Rate for Lapsed Period_Existing Rate for Remaining Period :&lt;br&gt; 1.For Lapsed Period :Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation). &lt;br&gt; 2.For Remaining Term: There will no interest rate change and contracted interest rate will be applicable for remaining term of TD.&lt;/li&gt;</li> </ul> <p>@param productId @return NN or EE or EN</p>	
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getContractModificationInterestRateFactor	<p>This method fetches the deposit contract modification recompute factor defined at product level, It could be one of the following</p> <ul style="list-style-type: none"> <li>&lt;li&gt;New Rate for Lapsed Period_New Rate for Remaining Period(NN) : &lt;br&gt; 1.For Lapse period : New Interest rate is fetched from pricing based on lapse no. of days.&lt;br&gt; Interest accrued till date is reversed and new interest is</li> </ul>	<p>This method fetches the deposit contract modification recompute factor defined at product level, It could be one of the following</p> <ul style="list-style-type: none"> <li>&lt;li&gt;New Rate for Lapsed Period_New Rate for Remaining Period(NN) : &lt;br&gt; 1.For Lapse period : New Interest rate is fetched from pricing based on lapse no. of days.&lt;br&gt; Interest accrued till date is reversed and new interest is computed based on new rate.&lt;br&gt; 2. For Remaining term : New interest rate will be fetched from pricing based on applicable facts.&lt;br&gt; New interest rate will be applicable for remaining term of TD&lt;/li&gt;</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>computed based on new rate.&lt;br&gt;  2. For Remanig term : New interest rate will be fetched from pricing based on applicable facts.&lt;br&gt; New interest rate will be applicable for remaining term of TD&lt;/li&gt;  &lt;li&gt;Existing Rate for Lapsed Period_New Rate for Remaining Period(EE) :&lt;br&gt; 1.For Lapsed Period: Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation).&lt;br&gt; 2.For Remaining Term:New interest rate will be fetched from pricing based on applicable facts.&lt;br&gt; New interest rate will be applicable for remaining term of TD.&lt;/li&gt; &lt;li&gt;Existing Rate for Lapsed Period_Existing Rate for Remaining Period :&lt;br&gt; 1.For Lapsed Period :Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation).&lt;br&gt; 2.For Remaining Term: There will no interest rate change and contracted interest rate will be applicable for remaining term of TD.&lt;/li&gt; &lt;/ul&gt;  @param productId @return NN or EE or EN</p>	<p>&lt;li&gt;Existing Rate for Lapsed Period_New Rate for Remaining Period(EE) :&lt;br&gt; 1.For Lapsed Period: Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation).&lt;br&gt; 2.For Remaining Term:New interest rate will be fetched from pricing based on applicable facts.&lt;br&gt; New interest rate will be applicable for remaining term of TD.&lt;/li&gt; &lt;li&gt;Existing Rate for Lapsed Period_Existing Rate for Remaining Period :&lt;br&gt; 1.For Lapsed Period :Interest rate till the lapse period will be based on contracted rate (at the time of deposit creation).&lt;br&gt; 2.For Remaining Term: There will no interest rate change and contracted interest rate will be applicable for remaining term of TD.&lt;/li&gt; &lt;/ul&gt;  @param productId @return NN or EE or EN</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsCarryForwardPayoutInstruction	This method returns true if CarryForward Payout Instruction is allowed else false. @param productId Term Deposit product id @return boolean	This method invokes executePreferenceOnline method of SettlementExecutionService for Batch Transactions. It reads the Settlement Response and updates the Transaction Detail @param sessionContext @param transactionDetail @param depositDTO @return @throws FatalException @throws FCRException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsCarryForwardPayoutInstruction	This method returns true if CarryForward Payout Instruction is allowed else false. @param productId Term Deposit product id @return boolean	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ul style="list-style-type: none"> <li>&lt;li&gt;Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process&lt;/li&gt;</li> <li>&lt;li&gt;Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ul style="list-style-type: none"> <li>&lt;li&gt;The {@link RenewalResponse} containing any amount which has to be renewed&lt;/li&gt;</li> <li>&lt;li&gt;The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}&lt;/li&gt;</li> <li>&lt;li&gt;The Deposit which is being Matured&lt;/li&gt;</li> </ul> </li> </ul> <p>The method reads the RenewalResponse. &lt;p&gt; If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances &lt;/p&gt; &lt;p&gt; If Deposit Returned from Renewal is null then</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link DepositMaturityService#matureDeposit } is invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsCarryForwardPayoutInstruction	This method returns true if CarryForward Payout Instruction is allowed else false. @param productId Term Deposit product id @return boolean	This method returns true if CarryForward Payout Instruction is allowed else false. @param productId Term Deposit product id @return boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method calls method from interest inquiry application service to calculate the interest for provided investment amount for the term using given interest rate, compounding and payout frequency. @param investmentSimulationRequestDTO {@link InvestmentSimulationRequestDTO} Object containing parameters required to calculate interest. @return {@link InvestmentSimulationResponseDTO} Object containing calculated interest. @see com.ofss.fc.app.adapter.td.ITDInterestAdapter#getInterestDetailsForInvestment(com.ofss.fc.app.td.dto.simulation.InvestmentSimulationRequestDTO)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will fetch all the product details for Term Deposit product class @param offerId @return Array of ProductDetailsForInterestSimulationDTO @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will simulate the maturity amount, principal balance and deposit term depending upon the input given and return the {@link TDInterestSimulationDTO} which will contain all the simulation details. @param investmentAmount @param maturityAmount @param depositTerm @param depositDate @param maturityDate @param interestRate @param payoutFrequency @param compoundingFrequency @param maturityFactor @param productId @return TDInterestSimulationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method fetches the Interest Details. This output DTO{@link TDInterestSimulationInterestRateDTO} contains {@link FactsRequiredForSimulationResponse} which stores the facts required for Simulation process, minimum deposit amount and minimum term specified for product, deposit frequency details specified at product level. @param sessionContext @param productId @param partyId @param currencyCode @return TDInterestSimulationInterestRateDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method calculates net rate applicable using base rate and userDiscretionaryMargin. @param simulationParameters {@link SimulationParameters} Holds all the input details to get rate @param pricingTicket Ticket from pricing module to get rate recommendations @return {@link InterestDetails} Object containing interest rate details
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method fetches product details from PMU. @param offerId Product code for which details to be fetched @param currencyCode Code of currency used @return {@link ProductAttributesDTO} Object containing product details. @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will fetch payout and compounding frequency for the product. @param simulationParameters {@link SimulationParameters} Object containing parameters required for fetching frequencies from product @return {@link DepositFrequencyDetailsDTO} Object containing interest payout and compounding frequency of the product @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will retrieve the facts that are required for simulation  Facts are required to obtain TD interest rates defined in Pricing module.  Product ids are input for this function as Product level and Bank level price policy charts will be searched for list of facts<> @param simulationFactInquiryRequest {@link SimulationFactInquiryRequest} Object containing parameters required for fetching facts @return Set of {@link FactDetails} objects containing details of facts required for simulation @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will fetch TD Interest rates from Pricing module  partyId, Ids of Products selected for simulation and Fact values will be passed as input, so that Pricing can evaluate Customer level, Product level and Bank level Price Policy charts for deriving interest rates.  @param simulationParameters {@link SimulationParameters} Holds all the input details to fetch interest details @return {@link RateOfferingResponse} Object containing pricing ticket @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method calculates net rate applicable using base rate and userDiscretionaryMargin. @param simulationParameters {@link SimulationParameters} Holds all the input details to get rate @param pricingTicket Ticket from pricing module to get rate recommendations @return {@link InterestDetails} Object containing interest rate details
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method populates simulation parameters for given parameters @param productCode TD product id @param depositTerm Term for deposit @param initialAmount Investment amount @param payoutFrequency Payout frequency for TD @return {@link SimulationParameters} Object populated with provided information





Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				#doPrincipalPayout} is invoked. </p> <p> If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked </p> In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked</li> </ol> Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method retrieves the InterestRuleKeyDTO from the PMU. @param productId The term deposit product id @return {@link InterestRuleKeyDTO} Object containing interest rule key parameters
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will open an account from existing account and new product. @param sessionContext @param accountId @param productId @return TermDepositAccountResponse @throws FatalException
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will fetch the required details for Interest Rate @param sessionContext @param productId @param partyId @param currencyCode @return TDInterestSimulationInterestRateResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method will validate the offer code  @param offerId @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method returns the ProductCode of the Offer @param offerID @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getProductCodeFromOffer	This method returns the ProductCode of the Offer @param offerID @return	This method returns the offerExpiry Date of the Offer @param offerID @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This method will fetch the details of TD account.   This inquiry will not give the joint holder details. @param sessionContext {@link SessionContext} @param accountId {@link TermDepositAccountKeyDTO} @return TransactionStatus {@link TransactionStatus} @throws FatalException {@link FatalException}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This service recomputes the statement generation dates based on provided {@link StatementFrequencyParameterDTO}.   @param sessionContext @param accountId @param newFrequency @return @throws FatalException @author abhishekbh @since R2.3
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This method validates the Statement parameters overridden at account level. @param termDepositAccount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This method assembles the statement parameters. @param termDepositAccount @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchStatementParameters	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO	This method fetches the product level default parameters for statement generation defined at product level @param productId @return StatementParametersDTO
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getCurrencyAttributes	This method returns currency attributes for the given product @param productId The Product for which product group id needs to be fetched @return String product id	This method will fetch all the product details for Term Deposit product class @param offerId @return Array of ProductDetailsForInterestSimulationDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getCurrencyAttributes	This method returns currency attributes for the given product  @param productId The Product for which product group id needs to be fetched @return String product id	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getMinimumTermIntEligibility(java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getCurrencyAttributes	This method returns currency attributes for the given product  @param productId The Product for which product group id needs to be fetched @return String product id	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#validateCurrencyDetails(java.lang.String, java.lang.String)
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getCurrencyAttributes	This method returns currency attributes for the given product  @param productId The Product for which product group id needs to be fetched @return String product id	This method returns currency attributes for the given product  @param productId The Product for which product group id needs to be fetched @return String product id
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	fetchOfferName	This method returns the offer name for given offer id @param offerId @return String offername	The method fetches the following parameters Branch Name, Product Name ,Party Name @param termDepositAccountDTO @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	validateBundleId	This method validates the bundle id passed. @param bundleId @return	This method validates the business policies  1. Call validateParties() {@link #validateParties(PartyAccountRelationshipDetails[])}  2. Call validateOpenDate() {@link #validateOpenDate(Date)}  3. Call validateMinorAccountStatus() {@link #validateMinorAccountStatus(TermDepositAccount)}  4. Call validateProduct() {@link #validateProduct(String)}  5. Call validateBranchCode() {@link #validateBranchCode(String)}  6. Call validateCurrency() {@link #validateCurrency(String)}  @param termDepositAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	validateBundleId	This method validates the bundle id passed. @param bundleId @return	This method will validate the offer code  @param offerId @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestOfferCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestOfferCode(java.lang.String)	This method is the Starting Point for Maturity Processing. It is Invoked from {@link DepositMaturityActionExecutor#executeAction}. At a Broad level the method performs the following actions <ol> <li>Fetch the Necessary Details, such as Instructions Maintained at Product which are required in the Decision Making Process</li> <li>Invoke {@link DepositMaturityService#processDepositMaturity}. This method returns a {@link MaturityResponse} object. The Response object contains the following Information <ol> <li>The {@link RenewalResponse} containing any amount which has to be renewed</li> <li>The {@link MaturityPayoutResponse} contains Principal, Interest Amount which is to be paid out and the {@link PayoutInstruction}</li> <li>The Deposit which is being Matured</li> </ol> </li> <li>The method reads the RenewalResponse. <p> If Not null then it invokes {@link DepositRenewalService#autoRenewDeposit}. If New Deposit received from DepositRenewalService is not Null then {@link DepositMaturityService#processPostMaturityRenewal} is invoked. This service will remove any existing Blocks or Liens from Matured Deposit and Carry forward the Lien or Block whichever is applicable to the New Deposit and accordingly update the Deposit Balances </p> <p> If Deposit Returned from Renewal is null then {@link DepositMaturityService#matureDeposit} is

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>invoked, which will update the Deposit Balances and mark the Deposit as Matured &lt;/p&gt; &lt;/li&gt; &lt;li&gt;If PayoutResponse object is present then the method determines the type of Payout. &lt;p&gt; If Principal Payout Instruction is Maintained then {@link #doPrincipalPayout} is invoked. &lt;/p&gt; &lt;p&gt; If PrincipalInterest Payout Instruction is Maintained then {@link #doPrincipalInterestPayout} is invoked &lt;/p&gt; In both the cases above if Payout fails then {@link DepositService#matureDeposit} is Invoked&lt;/li&gt; &lt;/ol&gt; Finally after the method exits, all the Deposit Balances and Deposit Status would have been marked as per the conditions present. Control will go back to the Maturity Action Executor. It will check if All the Deposit Balances have been zeroised in which case it will close the Deposit @param sessionContext @param deposit @throws FatalException</p>
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestOfferCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestOfferCode(java.lang.String)	This method reinvests the Deposit when Interest Payout Option is set to REINVEST @param interestPayoutInformation @param deposit @param renewalAmount @return Deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestOfferCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestOfferCode(java.lang.String)	This method prepares the InterestPayoutInformation @param applicationContext Application Context which will be set in batch process. @param actionKey {@link DepositBasedActionKey} object containing account and deposit id @param actionRequest {@link TDActionRequest} object containing td action to be performed @param dueDate due date for the action @param accountId id term deposit account @param depositId id of term deposit @return {@link InterestPayoutInformation} object containing information for interest payout processing

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getReinvestOfferCode	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getReinvestOfferCode(java.lang.String)	<p>This method reinvests the Deposit. It calculates the Value Date of the New Deposit and validates whether the Renewal Amount satisfies the Minimum Amount Required for the Product. If it is less, then the required amount is acquired from the Principal.   After the Deposit is Reinvested to a New account, the Nomination Details are carried forward to the New Account created {@link RenewalResponse} object is updated with Value Date and Renewal amount. </p><p> {@link MaturityResponse} object is updated with the RenewalResponse and the Deposit being matured, and returned </p> @param applicationContext @param deposit @param renewalAmount @param minRenewalAmount @return deposit Reinvested Deposit
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLockInTerm	This method fetches the lockIn term maintained at the offer level  @param offerId The offer for which lockInTerm needs to be fetched @return Duration lockInTerm	This method fetches the lockIn term maintained at the offer level  and Adds it to the Deposit Date to derive the lockIn Expiry Date for a deposit @param offerId Identifier to retrieve lockin term @param interestStartDate The date from which the interest calculation will start. @return lockIn Expiry Date
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getLockInTerm	This method fetches the lockIn term maintained at the offer level  @param offerId The offer for which lockInTerm needs to be fetched @return Duration lockInTerm	This method fetches the lockIn term maintained at the offer level  @param offerId The offer for which lockInTerm needs to be fetched @return Duration lockInTerm
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLockInTermEnabled	This method return true if lock in term has been enabled for the given offer. @param offerId The offer for which lockInTerm enabled or not that needs to be checked @return	The method populates {@link Deposit} object from {@link PayinTransactionDTO} @param {@link PayinTransactionDTO} @return {@link Deposit}



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isLockInTermEnabled	This method return true if lock in term has been enabled for the given offer. @param offerId The offer for which lockInTerm enabled or not that needs to be checked @return	This method return true if lock in term has been enabled for the given offer. @param offerId The offer for which lockInTerm enabled or not that needs to be checked @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getOfferExpiryDate	This method returns the offerExpiryDate of the Offer @param offerID @return	This method will validate the offer id  @param offerID @param productId
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getOfferExpiryDate	This method returns the offerExpiryDate of the Offer @param offerID @return	This method will validate the offer id  @param offerID @param productId
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isValidOfferBundleCombination	This method checks if the offer being used belongs to the corresponding bundle @param offerId @param bundleId @return	This method will check whether bundle id is null if bundle benefit applicable flag is true. @param termDepositAccount
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	isValidOfferBundleCombination	This method checks if the offer being used belongs to the corresponding bundle @param offerId @param bundleId @return	This method will validate the offer code  @param offerId @return boolean
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getBackdatedAcctOpeningPeriod		This method will validate the Opening date for the account  Opening date can not be less than the posting date @param accountOpenDate
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getBackdatedAcctOpeningPeriod		This method returns the backdated account opening allowed days. @param offerID @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAdditionalInterestPrincipalMaturity	This parameter specified whether interest has to be given for the difference of days between actual maturity date and the actual process date on which maturity action is executed, when the actual maturity date falls on a holiday. For instance if <ul style="list-style-type: none"> <li>Maturity Date : 10 Jun 2016 (Holiday)</li> <li>Next Working Day : 15 Jun 2016</li> </ul> <li>Hence the decision whether the interest given to the deposit should be till 10 Jun 2016 or 14 Jun 2016 will depend on this parameter. </li> @productCode @return	In case the maturity date falls on a holiday, it gets picked up for maturity processing on the next working day. In such a case whether the interest should be given for the number of days lapsed between the actual maturity date and the date on which the deposit gets actually picked up for maturity processing needs to be decided. The decision is made on the following : <ul style="list-style-type: none"> <li>The product level flag for additional interest on maturity is true.</li> <li>P+I instruction is maintained with the following settlement setup : <ul style="list-style-type: none"> <li>The settlement split type is PERCENT and split value is 100 (i.e single settlement mode)</li> <li>The settlement mode is NOT any flavor of CASA, TD or Loan account </li> </ul> </li> </ul> @param deposit @param dueDateFromAction @return
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAdditionalInterestPrincipalMaturity	This parameter specified whether interest has to be given for the difference of days between actual maturity date and the actual process date on which maturity action is executed, when the actual maturity date falls on a holiday. For instance if <ul style="list-style-type: none"> <li>Maturity Date : 10 Jun 2016 (Holiday)</li> <li>Next Working Day : 15 Jun 2016</li> </ul> <li>Hence the decision whether the interest given to the deposit should be till 10 Jun 2016 or 14 Jun 2016 will depend on this parameter. </li> @productCode @return	This method creates and returns an array of TDInterestDivisionDTO. On Deposit Maturity Interest has to be informed that the Deposit has matured and that Regular Interest is no longer to be applied. This is done by invoking a performInterestComputation method by passing a Zero Balance Base Amount and Interest Rate Zero for Regular Interest Division. Also If the Deposit is eligible for PMI Interest, then performInterestComputation on Interest is triggered for Deposit Uncollected Principal with Interest Division Post Maturity @param maturityProcessInformation @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDProductAdapter	getIsAdditionalInterestPrincipleMaturity	This parameter specified whether interest has to be given for the difference of days between actual maturity date and the actual process date on which maturity action is executed, when the actual maturity date falls on a holiday. For instance if <ul style="list-style-type: none"> <li>Maturity Date : 10 Jun 2016 (Holiday)</li> <li>Next Working Day : 15 Jun 2016</li> </ul> Hence the decision whether the interest given to the deposit should be till 10 Jun 2016 or 14 Jun 2016 will depend on this parameter. <ul style="list-style-type: none"> <li>@param productCode</li> <li>@return</li> </ul>	@see com.ofss.fc.app.adapter.td.ITDProductAdapter#getIsAddedPendingInterestToPrincipal(java.lang.String)
ILimitsSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\limits\ILimitsSettlementAdapter	doOnlineSettlement	This method calls the settlement module to do the online settlement of the deferred fees for the facility <ul style="list-style-type: none"> <li>@param eventCode</li> <li>@param productCode</li> <li>@param componentPreferenceDTO</li> <li>@param amount</li> <li>@param settlementValueDate</li> <li>@param narration</li> <li>@param messageType</li> <li>@param shortNarration</li> <li>@param sourceEventId</li> <li>@return FacilitySettlementResponseDTO</li> <li>@throws BusinessException</li> </ul>	This processes the deferred fees for a facility. First it calls the settlement module to process the deferred fees and then it will raise an accounting entry for the same. <ul style="list-style-type: none"> <li>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.</li> <li>The information is used by the system for authentication purposes.</li> <li>@fcb.param MI,com.ofss.fc.app.lcm.dto.limits.facility.creditfacility.DeferredFeesAmountDetailsDTO[], deferredFeesAmountDetailsDTOArray, This contains the amount details required for the settlement process</li> <li>@return {@link com.ofss.fc.service.response.TransactionStatus}</li> <li>@throws com.ofss.fc.infra.exception.FatalException</li> </ul> Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILimitsSettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\limits\ILimitsSettlementAdapter	doOnlineSettlement	This method calls the settlement module to do the online settlement of the   deferred fees for the facility @param eventCode @param productCode @param componentPreferenceDTO @param amount @param settlementValueDate @param narration @param messageType @param shortNarration @param sourceEventId @return FacilitySettlementResponseDTO @throws BusinessException	This method calls the settlement module to do the online settlement of the   deferred fees for the facility @param eventCode @param productCode @param componentPreferenceDTO @param amount @param settlementValueDate @param narration @param messageType @param shortNarration @param sourceEventId @return FacilitySettlementResponseDTO @throws BusinessException
IDDAELCMAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAELCMAdapter	inquireFacility	This method will be used to inquire the details of a facility from eLCM. The facility maintenance is done as part of eLCM. In the CASA module only the linkage between the account and facility is maintained. @param facilityId @param valueDate @param balanceCurrency @return	The method will be used to inquire the details of the facility attached for an account. At one point of time, an account can be linked to only facility. The method works as follows: <ul> <li>Call the method {@link DemandDepositAccountService#fetchOverdraftLimit} to get the overdraft limit details maintained as part of the CASA module. The attributes present in CASA module are represented by the domain entity OverdraftLimit.</li> <li>Populate the details in the OverdraftLimitDTO by calling the assembler method {@link OverdraftLimitAssembler#fromDomainObject}</li> <li>Call the method {@link IDDAELCMApplicationAdapter#inquireFacility} to get the facility details from eLCM. All the attributes of the facility which are required as part of inquiry and available in OverdraftLimitInquiryDTO is returned by this service.</li> </li> Set the required details in OverdraftLimitInquiryDTO by calling the method {@link OverdraftLimitAssembler#populateOverdraftLimitInquiryDTO}.</li> <li>Return the response OverdraftLimitInquiryResponse</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>@param sessionContext-This parameter captures the basic information about the user of the system</p> <p>@param accountId-Parameter for identifying Account.</p> <p>@throws FatalException @throws Business Exception DDA_ACCOUNT_NOT_PRESENT i.e) DDAccount can not be null or DDAccount does not exist with A/c No.</p>
IDDAELCM Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAELCMAdapter	setBalance	<p>This method resets the exposure maintained in &lt;b&gt;&lt;u&gt;elcm&lt;/u&gt;&lt;/b&gt; after every financial transaction.&lt;br&gt; @param inELCMUtilizationDetailsDTO @return</p>	<p>This method intimates eLCM about the balance utilization and deutilization.This method is called on every financial transaction.&lt;br&gt;This method works as follows : &lt;br&gt; &lt;pre&gt; Derive Principal balance for setBal() method. principalBalance = UtilizedOverdraftAmount() + TODAmount() + OverlineAmount() {The Overdraft, TOD and Overline amount are extracted from the input balance map} if(principalBalance &gt; 0) { &amp;lt;i&amp;gt;&amp;lt;/i&amp;gt;If it is available then pass it, else pass null.&amp;lt;/i&amp;gt; &amp;lt;i&amp;gt;&amp;lt;/i&amp;gt;In case of ODLimit utilization or Overline Utilization it will be passed to the ELCM service, If TOD is utilized, then it may or may not be derived.&amp;lt;/i&amp;gt; if(FCRThreadAttribute.AUTH_REQUIRED == false) { try { Call {@link IDDAELCMAdapter#setBal()} &amp;lt;i&amp;gt;&amp;lt;/i&amp;gt;params : Balance : principalBalance, forceAuth : 'N', AmountTag : 'PRINCIPAL'&amp;lt;/i&amp;gt; } catch(Override Exception e) { Call {@link IDDAELCMAdapter#setBal()} &amp;lt;i&amp;gt;&amp;lt;/i&amp;gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'PRINCIPAL'&amp;lt;/i&amp;gt; } } if(FCRThreadAttribute.AUTH_REQUIRED == true) { Call {@link IDDAELCMAdapter#setBal()} &amp;lt;i&amp;gt;&amp;lt;/i&amp;gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'PRINCIPAL'&amp;lt;/i&amp;gt; } } aufBalance = UtilizedAUFLimit() {UtilizedAUFLimit is extracted from the balance map}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre>if(aufBalance &gt; 0) { &amp;lt;i&amp;gt;No need to pass utilizationId&amp;lt;/i&amp;gt; Call {@link IDDAELCMAadapter#setBal() &amp;lt;i&amp;gt;params : Balance : principalBalance, forceAuth : 'Y', AmountTag : 'AUF'&amp;lt;/i&amp;gt; } &lt;/pre&gt; @param accountId @param dates @param balanceMap @param utilizationId</pre>
IBranchCommonSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchCommonSecurityAdapter	getUserProfileDetails		
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	addUnapprovedTransactionRequest	This method will add UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	addUnapprovedTransactionRequest	This method will add UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException	This method will add UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	addUnapprovedTransactionRequest	This method will add UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException	This method will update UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	fetchServiceName	This method returns the service name for the given transaction id. @param transactionId @return	This method returns the DMS monitoring serviceName. If the transaction is a normal transaction then the FCRThreadAttribute.DMS_SERVICE is same as FCRThreadAttribute.SERVICE. If the transaction is in 2FA then the serviceName is extracted using the transaction id and set in FCRThreadAttribute.DMS_SERVICE. If the transaction is in delay then the workitem is obtained using FCRThreadAttribute.DOMAIN_OBJECT and the service id is obtained from workitem and set in FCRThreadAttribute.DMS_SERVICE. @return
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	fetchServiceName	This method returns the service name for the given transaction id. @param transactionId @return	This method returns the service name for the given transaction id. @param transactionId @return
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	updateUnapprovedTransactionRequest	This method will update UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException	Method to persist UnApprovedTransactionRequest to DB. @throws FatalException
IUnApprovedTransactionRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IUnApprovedTransactionRequestAdapter	updateUnapprovedTransactionRequest	This method will update UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException	This method will update UnapprovedTransactionRequest to DB. @param sessionContext @param unApprovedTransactionRequestDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IJournalEnrichmentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IJournalEnrichmentAdapter	enrich		This method will try to insert or update transaction log item for given transaction reference.   The system will check existence of teller log item using the key sent in the input FinancialStatementItem.   It will then apply following rules before updating the existing record <ul> <li>If the transaction amount's currencies are same, then item will highest transaction value gets persisted <li>If the transaction amount's currencies are same and monetary values are also same, then one with lowest subsequence number gets persisted <li>If the transaction amount's currencies are different, then one with lowest subsequence number gets persisted. </ul> @param repository @param incomingTransactionLogItem @param existingTransactionLogItem @param module
IJournalEnrichmentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IJournalEnrichmentAdapter	enrich		Enriches the journal item by adding extra information that is not present in the {@link FinancialStatementItem}. Moudle specific enrichment needs t be done at times depending on the type of transaction.  For ex., there is a need to add account status at time of transaction to the journal @param journalItemDTO @param incomingTransactionLogItem @param module
IJournalEnrichmentAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IJournalEnrichmentAdapter	enrich		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IReasonCodesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IReasonCodesAdapter	checkAuthorization	This method fetches the severity and workflow depending on the reasonCode specified. It attaches the fetched workflow to the list which has to be sent. Before adding the workflow to the list it will first check, whether workflow for the same reasonCode exists or not, if not it will simply add the fetched workflow to the list. @fcb.param MI,String,channelId, ,Indicates the channel Id. @fcb.param MI,SessionContext,sessionContext, ,Indicates the sessionContext. @throws FatalException.	This method fetches the severity and workflow depending on the reasonCode specified. It attaches the fetched workflow to the list which has to be sent. Before adding the workflow to the list it will first check, whether workflow for the same reasonCode exists or not, if not it will simply add the fetched workflow to the list. @fcb.param MI,String,channelId, ,Indicates the channel Id. @fcb.param MI,SessionContext,sessionContext, ,Indicates the sessionContext. @throws FatalException
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method retrieves the verification checklist for an application. It will do the following 1. Fetch the Verification Checklist ID linked with the Application ID 2. Call the method fetchVerificationChecklist() in VerificationApplicationService 3. Copy the common services response object contents to the origination response object @fcb.param MI SessionContext sessionContext, denotes the session context @fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @return OriginationChecklistInquiryResponse , The response class containing the checklist DTO @throws FatalException  

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method checks if the submission has already been verified before. This is done by checking the linkage tables maintained in origination @param String submissionId @return Boolean
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext	This method checks if the application has already been verified before. This is done by checking the linkage tables maintained in origination @param String applicationId, denotes the applicationId to be checked @return Boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param verificationChecklistId	
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method begins the process of verification @param sessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @param application denotes the application domain object @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	<p>This method takes in the verified checklist ID and returns the &lt;br&gt;{@link VerificationChecklistResponse} which contains the verification checklist and &lt;br&gt; associated Verifications. Caller needs to know the verification checklist ID to pass in &lt;br&gt; to the method's input.&lt;br&gt;&lt;br&gt; For example, when Origination needs to inquire verification checklist for verify &lt;br&gt; application data event, it should pass the verification checklist ID stored in Origination &lt;br&gt; module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId</p>	<p>This method will take in an application ID and return a boolean value indicate &lt;br&gt; whether verification checklist has been created for this application. This method will be used &lt;br&gt; by the Worklist to check if verification workitem has to be spawned for the &lt;br&gt; application. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt;@fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @return OriginationVerificationRequiredResponse , The response class containing a flag that indicates whether a worklist item has to be spawned for the application @throws FatalException &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	<p>This method takes in the verified checklist ID and returns the &lt;br&gt;{@link VerificationChecklistResponse} which contains the verification checklist and &lt;br&gt; associated Verifications. Caller needs to know the verification checklist ID to pass in &lt;br&gt; to the method's input.&lt;br&gt;&lt;br&gt; For example, when Origination needs to inquire verification checklist for verify &lt;br&gt; application data event, it should pass the verification checklist ID stored in Origination &lt;br&gt; module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId</p>	<p>This method will take an applicationId and return a boolean flag indicates &lt;br&gt; whether the Verification is done. It will call the service provided in &lt;br&gt; VerificationApplicationService to get the indicator. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt;@fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @return OriginationVerificationIndicatorResponse , The response class containing a boolean flag indicates whether the Verification is done @throws FatalException &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method will take a submissionId and return a boolean flag indicates   whether the Verification is done. It will call the service provided in   VerificationApplicationService to get the indicator. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @return OriginationVerificationIndicatorResponse , The response class containing a boolean flag indicates   whether the Verification is done for the submission @throws FatalException  

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method will fetch the verification checklist ID stored in origination for the given application, it is used by the UI to fetch the verification checklist from common services module. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @fcb.param MI String application Id @return OriginationVerificationIndicatorResponse the response contains the checklist Id @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @throws FatalException 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method will be invoked by the event triggered for verification from credit assessment stage. The VerificationInputDTO contains submission to be verified. The verification policy will be executed for each application of the submission. If checklist already exists for the application, the verifications will be appended to it. Otherwise new checklist will be created. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @fcb.param MI VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @throws FatalException  



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	<p>This method takes in the verified checklist ID and returns the &lt;br&gt;{@link VerificationChecklistResponse} which contains the verification checklist and &lt;br&gt; associated Verifications. Caller needs to know the verification checklist ID to pass in &lt;br&gt; to the method's input.&lt;br&gt;&lt;br&gt; For example, when Origination needs to inquire verification checklist for verify &lt;br&gt; application data event, it should pass the verification checklist ID stored in Origination &lt;br&gt; module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId</p>	<p>This method will submit the Application Verification task. If there are any waived verifications, approval workflow is initiated. The application verified date is updated. @param SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param String applicationId, denotes the application id @param ApplicationApprovalDataDTO applicationApprovalDataDTO , denotes the application Approval Data DTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#WAIVED_VERIFICATION_APPROVAL_REQUIRED} : FC_OR_AV_0005 &lt;br&gt; This overridable exception occurs if:&lt;br&gt; There are any waived verification(s).The same will be sent for approval&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PENDING_VERIFICATIONNS_SUBMIT} : FC_OR_AV_0006 &lt;br&gt; This exception occurs if there are pending verifications. The task cannot be submitted in that case. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method calls publishOriginationActivity from OriginationEventApplicationService which logs and processes events related to Verification Activity.It will internally call registerActivity from AlertsNotificationHandler which performs eventProcessing.This alert will be called upon successful verification under given application. @param sessionContext @param checklistId @param applicationId
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param verificationChecklistId	
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationChecklist	This method takes in the verified checklist ID and returns the  {@link VerificationChecklistResponse} which contains the verification checklist and   associated Verifications. Caller needs to know the verification checklist ID to pass in   to the method's input.   For example, when Origination needs to inquire verification checklist for verify   application data event, it should pass the verification checklist ID stored in Origination   module to the method to get the verification checklist. @param sessionContext @param verificationChecklistId	This method calls the fetchVerificationChecklist method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId @return VerificationChecklistResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklistGroup	<p>This method creates new verification checklist group and returns the checklist group ID &lt;br&gt; to the caller. The caller need to keep track of the checklist group ID in order to add a &lt;br&gt; new verification checklist to that group. This is optional since verification checklist can &lt;br&gt; stand alone. If the caller need to link many checklist together, it need to first create &lt;br&gt; the checklist.&lt;br&gt; &lt;br&gt; For example, in order to create checklists to verify Applications in Origination, the caller &lt;br&gt; need to first create a checklist group associated with the Submission. Then it call the &lt;br&gt; addChecklistToGroup method with the checklist group ID as one of the input. &lt;br&gt; VerificationApplicationService will check for existing verifications in the checklist group &lt;br&gt; and link them to the new added checklist.</p> <p>@param sessionContext</p>	<p>This method will fetch the ChecklistGroupID from common services @param sessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @return String checklistGroupId, denotes the checklist group id @throws Exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklistGroup	<p>This method creates new verification checklist group and returns the checklist group ID &lt;br&gt; to the caller. The caller need to keep track of the checklist group ID in order to add a &lt;br&gt; new verification checklist to that group. This is optional since verification checklist can &lt;br&gt; stand alone. If the caller need to link many checklist together, it need to first create &lt;br&gt; the checklist.&lt;br&gt; &lt;br&gt; For example, in order to create checklists to verify Applications in Origination, the caller &lt;br&gt; need to first create a checklist group associated with the Submission. Then it call the &lt;br&gt; addChecklistToGroup method with the checklist group ID as one of the input. &lt;br&gt; VerificationApplicationService will check for existing verifications in the checklist group &lt;br&gt; and link them to the new added checklist.</p> <p>@param sessionContext</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklistGroup	<p>This method creates new verification checklist group and returns the checklist group ID &lt;br&gt; to the caller. The caller need to keep track of the checklist group ID in order to add a &lt;br&gt; new verification checklist to that group. This is optional since verification checklist can &lt;br&gt; stand alone. If the caller need to link many checklist together, it need to first create &lt;br&gt; the checklist.&lt;br&gt; &lt;br&gt; For example, in order to create checklists to verify Applications in Origination, the caller &lt;br&gt; need to first create a checklist group associated with the Submission. Then it call the &lt;br&gt; addChecklistToGroup method with the checklist group ID as one of the input. &lt;br&gt; VerificationApplicationService will check for existing verifications in the checklist group &lt;br&gt; and link them to the new added checklist.</p> <p>@param sessionContext</p>	<p>This method calls the createChecklistGroup method in VerificationApplication service in Common services module. @param SessionContext @return VerificationChecklistGroupCreationResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; <a href="#">@link CreateVerificationChecklistResponse</a> which contains the checklist ID to &lt;br&gt; This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	<p>This method will fetch the ChecklistGroupID from common services @param sessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @return String checklistGroupId, denotes the checklist group id @throws Exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; (@link CreateVerificationChecklistResponse) which contains the checklist ID to &lt;br&gt; This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	<p>This method creates the checklist i.e inserts checklist into the DB @param sessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param verificationChecklistDTO denotes the checklist to be inserted into the DB @return String verificationChecklistId , denotes the checklist id @throws Exception</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; @link CreateVerificationChecklistResponse) which contains the checklist ID to &lt;br&gt; This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; @link CreateVerificationChecklistResponse) which contains the checklist ID to &lt;br&gt;This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; <a href="#">@link CreateVerificationChecklistResponse</a> which contains the checklist ID to &lt;br&gt; This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	<p>This method calls the createChecklist method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistDTO @return VerificationChecklistCreationResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	createChecklist	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. createChecklist will build the checklist and linkages from the &lt;br&gt; input and persist the data to the database. It will also return the &lt;br&gt; {@link CreateVerificationChecklistResponse} which contains the checklist ID to &lt;br&gt; This method will call createChecklistGroup to create checklist group before adding the checklist the caller.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. createChecklist will return &lt;br&gt; the verification checklist ID so that Origination can store this value for inquiry. @param sessionContext @param checklistDTO</p>	<p>This method calls the createChecklistGroup method in VerificationApplication service in Common services module. @param SessionContext @return VerificationChecklistGroupCreationResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {@link CreateVerificationChecklistResponse} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	<p>This method will be invoked by the process for deposit application verifications. The VerificationInputDTO contains the submissionId or the applicationId to be verified. @param Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException &lt;br&gt; @exception &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_NOT_FOUND} : FC_OR_DE_0055 &lt;br&gt; error occurs when:&lt;br&gt; The details for the submission identifier were not found or the submission identifier is incorrect.&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#CHECKLIST_APPLICATION_NOT_FOUND} : FC_OR_AV_0001 &lt;br&gt; This exception is thrown if there is no verification checklist for the application &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#CHECKLISTGROUP_SUBMISSION_NOT_FOUND} : FC_OR_AV_0002 &lt;br&gt; This exception is thrown if there is no verification checklist for the submission &lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#SUBMISSION_ALREADY_VERIFIED} : FC_OR_AV_0003 &lt;br&gt; This exception is thrown when the submission is already verified &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p data-bbox="1295 300 1950 427">&lt;br&gt; {@link OriginationErrorConstants#APPLICATION_ALREADY_VERIFIED} : FC_OR_AV_0004 &lt;br&gt; This exception is thrown when the application is already verified</p> <p data-bbox="1295 427 1950 621">&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#ALL_PRODUCT_GROUPS_WITHDRAWN} : ALL_PRODUCT_GROUPS_WITHDRAWN &lt;br&gt; This exception is thrown when all the product groups in the application have been withdrawn &lt;br&gt; &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {&lt;br&gt; @link CreateVerificationChecklistResponse} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	<p>This method creates the checklist i.e inserts checklist into the DB @param sessionContext This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param verificationChecklistDTO denotes the checklist to be inserted into the DB @return String verificationChecklistId , denotes the checklist id @throws Exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {<a href="#">@link CreateVerificationChecklistResponse</a>} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	<p>This method will be invoked by the event triggered for verification from credit assessment stage. The VerificationInputDTO contains submission to be verified. The list of verifications to be done for the application is obtained from common service module. To this list, the extra verifications obtained from Credit Assessment stage are added. The final verification checklist is then saved. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @fcb.param MI OriginationVerificationDTO[] - the list of extra verifications obtained from credit assessment stage @return {<a href="#">@link com.ofss.fc.service.response.TransactionStatus</a>} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException&lt;br&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {<a href="#">@link CreateVerificationChecklistResponse</a>} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	<p>This method will be invoked by the event triggered for verification from credit assessment stage. The VerificationInputDTO contains submission to be verified. The verification policy will be executed for each application of the submission. If checklist already exists for the application, the verifications will be appended to it. Otherwise new checklist will be created. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @return {<a href="#">@link com.ofss.fc.service.response.TransactionStatus</a>} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {@link CreateVerificationChecklistResponse} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	addChecklistToGroup	<p>This method creates a new Verification Checklist in the system. A list of verifications is &lt;br&gt; required as an input. addChecklistToGroup will build the checklist and linkages from &lt;br&gt; the input and persist the data to the database. It will also return the &lt;br&gt; {@link CreateVerificationChecklistResponse} which contains the checklist ID to the &lt;br&gt; caller.&lt;br&gt; In addition, the method also requires the checklist group ID to link the verification.&lt;br&gt; &lt;br&gt; For example, submitting new created application in Origination will trigger an event to &lt;br&gt; build the verification checklist. Origination will then pass this checklist to &lt;br&gt; VerificationApplicationService to persist in to the Database. &lt;br&gt; VerificationApplicationService also link the new checklist to the existing group. &lt;br&gt; addVerificationChecklistToGroup will also return the verification checklist ID so that &lt;br&gt; Origination can store this value for inquiry. @param sessionContext @param checklist @param verificationChecklistGroupId</p>	<p>This method calls the addChecklistToGroup method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistDTO, String @return VerificationChecklistCreationResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	This methods updates the verification checklist. All the existing verifications in the   verification checklist will be updated. The updated information includes user comments,   linked documents and new verification added by user. All the data will be updated in   one function call. The method requires the checklist ID as an input. It will load the   checklist and update all the verifications and linkages in the checklist.    For example, a banker adds some uploaded documents to some verification in a   checklist. He or she also add some comments and an ah-hoc verification. All the   information will be captured in the checklist and send to the VerificationApplicationService.  @param sessionContext @param checklist	This method will update verification checklist that linked to an application. It   will call the VerificationApplicationService in Common Service module to   persist the data. 1. Convert from OriginationChecklistDTO to CsServices ChecklistDTO 2. call update method of common services 3. Once all the verifications in the application are done, the verified date is updated in the application checklist linkage table in Origination @param SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @param OriginationChecklistDTO, denotes the checklist that contains verifications to be updated @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @throws FatalException @exception   {@link OriginationErrorConstants#CHECKLIST_APPLICATION_NOT_FOUND} : FC_OR_AV_0001   This exception is thrown if there is no verification checklist for the application  

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	<p>This method will call the corresponding method in &lt;br&gt; VerificationApplicationService to update the verification checklist status. The &lt;br&gt; status could be either VALID or INVALID. If the status is VALID, the statuses &lt;br&gt; of all verification associated with the checklist will be valid. Otherwise, the &lt;br&gt; statuses will be invalid even though the verifications status are SUCCESSFUL.&lt;br&gt;The method will then reset all the statuses of verifications.&lt;br&gt; &lt;br&gt; For example, when an application is rolled back in Origination, the system will &lt;br&gt; set the status of the checklist to INVALID. Thus the whole application &lt;br&gt; verifications will become INVALID and verifications need to be redone. As per the R2.3 requirements, the verifications should be moved to the respective history tables</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI String verificationChecklistId, denotes the checklist id whose status has to be updated</p> <p>@return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException</p> <p>&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	<p>This method will submit the Application Verification task. If there are any waived verifications, approval workflow is initiated. The application verified date is updated. @param SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param String applicationId, denotes the application id @param ApplicationApprovalDataDTO applicationApprovalDataDTO , denotes the application Approval Data DTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @task</p> <p>OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException</p> <p>@exception &lt;br&gt; {@link OriginationErrorConstants#WAIVED_VERIFICATION_APPROVAL_REQUIRED} : FC_OR_AV_0005 &lt;br&gt; This overridable exeption occurs if:&lt;br&gt; There are any waived verification(s).The same will be sent for approval&lt;br&gt; &lt;br&gt; {@link OriginationErrorConstants#PENDING_VERIFICATIONNS_SUBMIT} : FC_OR_AV_0006 &lt;br&gt; This exception occurs if there are pending verifications. The task cannot be submitted in that case. &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	<p>This method calls the updateChecklist method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistDTO @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklist	<p>This methods updates the verification checklist. All the existing verifications in the &lt;br&gt; verification checklist will be updated. The updated information includes user comments, &lt;br&gt; linked documents and new verification added by user. All the data will be updated in &lt;br&gt; one function call. The method requires the checklist ID as an input. It will load the &lt;br&gt; checklist and update all the verifications and linkages in the checklist.&lt;br&gt; &lt;br&gt; For example, a banker adds some uploaded documents to some verification in a &lt;br&gt; checklist. He or she also add some comments and an ah-hoc verification. All the &lt;br&gt; information will be captured in the checklist and send to the VerificationApplicationService.</p> <p>@param sessionContext @param checklist</p>	<p>This method calls the updateChecklistStatus method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId, VerificationChecklistStatusType @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistIndicator	<p>This method will take in a verification checklist ID. It fetches all the linked &lt;br&gt; VerificationDetail of the entity. It will check whether all the verifications are successful, &lt;br&gt; set up the flag for {@link VerificationIndicatorResponse} accordingly and return the &lt;br&gt; result.&lt;br&gt; &lt;br&gt; For example, in the Create Offer task of an Application, the Verification Indicator for &lt;br&gt; the application will be shown up on the Create Offer screen. This indicator will be fetch &lt;br&gt; from the Checklist ID that stored in Origination. VerificationApplicationService will first &lt;br&gt; get this ID from the caller and look for all the verification under that checklist. If all &lt;br&gt; the status are Successful, the return value will be true. Otherwise, false will be returned. @param sessionContext @param linkedEntityReference @param verificationChecklistId</p>	<p>This method will update verification checklist that linked to an application. It &lt;br&gt; will call the VerificationApplicationService in Common Service module to &lt;br&gt; persist the data. 1. Convert from OriginationChecklistDTO to CsServices ChecklistDTO 2. call update method of common services 3. Once all the verifications in the application are done, the verified date is updated in the application checklist linkage table in Origination @param SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @param OriginationChecklistDTO, denotes the checklist that contains verifications to be updated @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException @exception &lt;br&gt; {@link OriginationErrorConstants#CHECKLIST_APPLICATION_NOT_FOUND} : FC_OR_AV_0001 &lt;br&gt; This exception is thrown if there is no verification checklist for the application &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistIndicator	<p>This method will take in a verification checklist ID. It fetches all the linked &lt;br&gt; VerificationDetail of the entity. It will check whether all the verifications are successful, &lt;br&gt; set up the flag for {@link VerificationIndicatorResponse} accordingly and return the &lt;br&gt; result.&lt;br&gt; &lt;br&gt; For example, in the Create Offer task of an Application, the Verification Indicator for &lt;br&gt; the application will be shown up on the Create Offer screen. This indicator will be fetch &lt;br&gt; from the Checklist ID that stored in Origination. VerificationApplicationService will first &lt;br&gt; get this ID from the caller and look for all the verification under that checklist. If all &lt;br&gt; the status are Successful, the return value will be true. Otherwise, false will be returned. @param sessionContext @param linkedEntityReference @param verificationChecklistId</p>	<p>This method will take an applicationId and return a boolean flag indicates &lt;br&gt; whether the Verification is done. It will call the service provided in &lt;br&gt; VerificationApplicationService to get the indicator. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @return OriginationVerificationIndicatorResponse , The response class containing a boolean flag indicates whether the Verification is done @throws FatalException &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistIndicator	<p>This method will take in a verification checklist ID. It fetches all the linked &lt;br&gt; VerificationDetail of the entity. It will check whether all the verifications are successful, &lt;br&gt; set up the flag for {@link VerificationIndicatorResponse} accordingly and return the &lt;br&gt; result.&lt;br&gt; &lt;br&gt; For example, in the Create Offer task of an Application, the Verification Indicator for &lt;br&gt; the application will be shown up on the Create Offer screen. This indicator will be fetch &lt;br&gt; from the Checklist ID that stored in Origination. VerificationApplicationService will first &lt;br&gt; get this ID from the caller and look for all the verification under that checklist. If all &lt;br&gt; the status are Successful, the return value will be true. Otherwise, false will be returned. @param sessionContext @param linkedEntityReference @param verificationChecklistId</p>	<p>This method will append a list of verification to a verification checklist. The new &lt;br&gt; verification will be added to the system. New verification linkages will also be added &lt;br&gt; to the verification checklist.&lt;br&gt; &lt;br&gt; For example, in credit assessment stage in Origination, some additional verifications &lt;br&gt; need to be done for the primary applicant. In this case, the event from the credit &lt;br&gt; assessment stage will be triggered and appendVerificationToChecklist will be called. &lt;br&gt;The method will first add all the verifications and then create the linkages for the&lt;br&gt; verification checklist. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,OriginationVerificationDTO[],verificationDTO, verifications to be added to existing checklist @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws com.ofss.fc.infra.exception.FatalException &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistIndicator	<p>This method will take in a verification checklist ID. It fetches all the linked &lt;br&gt; VerificationDetail of the entity. It will check whether all the verifications are successful, &lt;br&gt; set up the flag for {@link VerificationIndicatorResponse} accordingly and return the &lt;br&gt; result.&lt;br&gt; &lt;br&gt; For example, in the Create Offer task of an Application, the Verification Indicator for &lt;br&gt; the application will be shown up on the Create Offer screen. This indicator will be fetch &lt;br&gt; from the Checklist ID that stored in Origination. VerificationApplicationService will first &lt;br&gt; get this ID from the caller and look for all the verification under that checklist. If all &lt;br&gt; the status are Successful, the return value will be true. Otherwise, false will be returned. @param sessionContext @param linkedEntityReference @param verificationChecklistId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistIndicator	<p>This method will take in a verification checklist ID. It fetches all the linked &lt;br&gt; VerificationDetail of the entity. It will check whether all the verifications are successful, &lt;br&gt; set up the flag for {@link VerificationIndicatorResponse} accordingly and return the &lt;br&gt; result.&lt;br&gt; &lt;br&gt; For example, in the Create Offer task of an Application, the Verification Indicator for &lt;br&gt; the application will be shown up on the Create Offer screen. This indicator will be fetch &lt;br&gt; from the Checklist ID that stored in Origination. VerificationApplicationService will first &lt;br&gt; get this ID from the caller and look for all the verification under that checklist. If all &lt;br&gt; the status are Successful, the return value will be true. Otherwise, false will be returned. @param sessionContext @param linkedEntityReference @param verificationChecklistId</p>	<p>This method calls the fetchChecklistIndicator method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId @return VerificationIndicatorResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistGroupIndicator		This method will take a submissionId and return a boolean flag indicates   whether the Verification is done. It will call the service provided in   VerificationApplicationService to get the indicator. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI String applicationId, denotes the application id @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @return OriginationVerificationIndicatorResponse , The response class containing a boolean flag indicates   whether the Verification is done for the submission @throws FatalException  
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistGroupIndicator		
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchChecklistGroupIndicator		This method calls the fetchChecklistGroupIndicator method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistGroupId @return VerificationIndicatorResponse



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklistStatus	<p>This method will update the verification checklist status. The status could be either &lt;br&gt; VALID or INVALID. If the status is VALID, the statuses of all verification associated &lt;br&gt; with the checklist will be valid. Otherwise, the statuses will be invalid even though &lt;br&gt; the verifications status are SUCCESSFUL. The method will then reset all the statuses &lt;br&gt; of verifications.&lt;br&gt; &lt;br&gt; For example, when an application is rollback in Origination, the system will set the &lt;br&gt; status of the checklist to INVALID. Thus the whole application Verifications will &lt;br&gt; become INVALID and verifications need to be redone.</p> <p>@param sessionContext @param verificationChecklistId @param status</p>	<p>This method will call the corresponding method in &lt;br&gt; VerificationApplicationService to update the verification checklist status. The &lt;br&gt; status could be either VALID or INVALID. If the status is VALID, the statuses &lt;br&gt; of all verification associated with the checklist will be valid. Otherwise, the &lt;br&gt; statuses will be invalid even though the verifications status are SUCCESSFUL.&lt;br&gt;The method will then reset all the statuses of verifications.&lt;br&gt; &lt;br&gt; For example, when an application is rolled back in Origination, the system will &lt;br&gt; set the status of the checklist to INVALID. Thus the whole application &lt;br&gt; verifications will become INVALID and verifications need to be redone. As per the R2.3 requirements, the verifications should be moved to the respective history tables @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI String verificationChecklistId, denotes the checklist id whose status has to be updated @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklistStatus	<p>This method will update the verification checklist status. The status could be either &lt;br&gt; VALID or INVALID. If the status is VALID, the statuses of all verification associated &lt;br&gt; with the checklist will be valid. Otherwise, the statuses will be invalid even though &lt;br&gt; the verifications status are SUCCESSFUL. The method will then reset all the statuses &lt;br&gt; of verifications.&lt;br&gt; &lt;br&gt; For example, when an application is rollback in Origination, the system will set the &lt;br&gt; status of the checklist to INVALID. Thus the whole application Verifications will &lt;br&gt; become INVALID and verifications need to be redone.</p> <p>@param sessionContext @param verificationChecklistId @param status</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	updateChecklistStatus	<p>This method will update the verification checklist status. The status could be either &lt;br&gt; VALID or INVALID. If the status is VALID, the statuses of all verification associated &lt;br&gt; with the checklist will be valid. Otherwise, the statuses will be invalid even though &lt;br&gt; the verifications status are SUCCESSFUL. The method will then reset all the statuses &lt;br&gt; of verifications.&lt;br&gt; &lt;br&gt; For example, when an application is rollback in Origination, the system will set the &lt;br&gt; status of the checklist to INVALID. Thus the whole application Verifications will &lt;br&gt; become INVALID and verifications need to be redone.</p> <p>@param sessionContext @param verificationChecklistId @param status</p>	<p>This method calls the updateChecklistStatus method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId, VerificationChecklistStatusType @return TransactionStatus</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param sessionContext @param verificationContext @param effectiveDate @param facts	This method is used to validate the attributes for a verification checklist and the verifications and entity linkages contained in it @param originationChecklistDTO @return
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param sessionContext @param verificationContext @param	This method will construct the verifications for all applicants in the application. It will give back an arraylist that will contain array of VerificationDTO's @param sessionContext @param application @return ArrayList<VerificationDTO[]> @throws Exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			effectiveDate @param facts	
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param sessionContext @param verificationContext @param effectiveDate @param facts	This method will construct the verifications for all applicants in the application. It will give back an arraylist that will contain array of VerificationDTO's @param sessionContext @param application @return ArrayList<VerificationDTO[]> @throws Exception
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	This method will take in a verification context which has the event ID, domain and   domain category. It will call the VerificationPolicyApplicationService to get the rule ID.   From the effective date, facts and rule ID, it will call the   RuleExecutionApplicationService to get the list of Verification Policy and/or   Verification Types. The Verification Policy will be used to get the Types. The final   Verification Types list will be returned to the caller. @param	This method constructs the verification checklist DTO for verificationApplication common service from the OriginationChceklistDTO. @param originationChecklistDTO - origination checklistDTo @return VerificationChecklistDTO - verifictaion checklistDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>sessionContext @param  verificationContext @param  effectiveDate @param facts</p>	
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	<p>This method will take in a verification context which has the event ID, domain and &lt;br&gt; domain category. It will call the VerificationPolicyApplicationService to get the rule ID. &lt;br&gt; From the effective date, facts and rule ID, it will call the &lt;br&gt; RuleExecutionApplicationService to get the list of Verification Policy and/or &lt;br&gt; Verification Types. The Verification Policy will be used to get the Types. The final &lt;br&gt; Verification Types list will be returned to the caller. @param sessionContext @param verificationContext @param effectiveDate @param facts</p>	<p>This method will be invoked by the event triggered for verification from credit assessment stage. The VerificationInputDTO contains submission to be verified. The list of verifications to be done for the application is obtained from common service module. To this list, the extra verifications obtained from Credit Assessment stage are added. The final verification checklist is then saved. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @fcb.param MI OriginationVerificationDTO[] - the list of extra verifications obtained from credit assessment stage  @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getVerifications	<p>This method will take in a verification context which has the event ID, domain and &lt;br&gt; domain category. It will call the VerificationPolicyApplicationService to get the rule ID. &lt;br&gt; From the effective date, facts and rule ID, it will call the &lt;br&gt; RuleExecutionApplicationService to get the list of Verification Policy and/or &lt;br&gt; Verification Types. The Verification Policy will be used to get the Types. The final &lt;br&gt; Verification Types list will be returned to the caller. @param sessionContext @param verificationContext @param effectiveDate @param facts</p>	<p>This method calls the getVerifications method in VerificationApplication service in Common services module. @param SessionContext, VerificationContextDTO Effective Date, EntityLinkageLevelType, FactInformationDTO[] @return VerificationTypeResponse</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	appendVerificationToChecklist	This method will append a list of verification to a verification checklist. The new   verification will be added to the system. New verification linkages will also be added   to the verification checklist.    For example, in credit assessment stage in Origination, some additional verifications   need to be done for the primary applicant. In this case, the event from the credit   assessment stage will be triggered and appendVerificationToChecklist will be called.  The method will first add all the verifications and then create the linkages for the  verification checklist. @param sessionContext @param verifications @param verificationChecklistId	This method will append a list of verification to a verification checklist. The new   verification will be added to the system. New verification linkages will also be added   to the verification checklist.    For example, in credit assessment stage in Origination, some additional verifications   need to be done for the primary applicant. In this case, the event from the credit   assessment stage will be triggered and appendVerificationToChecklist will be called.  The method will first add all the verifications and then create the linkages for the  verification checklist. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.  @fcb.param MI,OriginationVerificationDTO[],verificationDTO, verifications to be added to existing checklist @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.  @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION  @throws com.ofss.fc.infra.exception.FatalException  



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	appendVerificationToChecklist	<p>This method will append a list of verification to a verification checklist. The new &lt;br&gt; verification will be added to the system. New verification linkages will also be added &lt;br&gt; to the verification checklist.&lt;br&gt; &lt;br&gt; For example, in credit assessment stage in Origination, some additional verifications &lt;br&gt; need to be done for the primary applicant. In this case, the event from the credit &lt;br&gt; assessment stage will be triggered and appendVerificationToChecklist will be called. &lt;br&gt;The method will first add all the verifications and then create the linkages for the&lt;br&gt; verification checklist. @param sessionContext @param verifications @param verificationChecklistId</p>	<p>This method will be invoked by the event triggered for verification from credit assessment stage. The VerificationInputDTO contains submission to be verified. The verification policy will be executed for each application of the submission. If checklist already exists for the application, the verifications will be appended to it. Otherwise new checklist will be created. @fcb.param MI Sessioncontext sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI VerificationInputDTO verificationInputDTO, This input DTO is passed on by the event to this service. It contains the application/submission that has to be verified and the event that fired the request @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @throws FatalException&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	appendVerificationToChecklist	This method will append a list of verification to a verification checklist. The new   verification will be added to the system. New verification linkages will also be added   to the verification checklist.    For example, in credit assessment stage in Origination, some additional verifications   need to be done for the primary applicant. In this case, the event from the credit   assessment stage will be triggered and appendVerificationToChecklist will be called.  The method will first add all the verifications and then create the linkages for the  verification checklist. @param sessionContext @param verifications @param verificationChecklistId	This method calls the appendVerificationToChecklist method in VerificationApplication service in Common services module. @param SessionContext, VerificationDTO[], String @return TransactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationHistory	<p>This method will take in a verification ID and return the complete history of the &lt;br&gt; verification.&lt;br&gt; &lt;br&gt; For example, the banker might want to view the history of some verification he is &lt;br&gt; working on. On select the verification (or verification ID), the system will return the &lt;br&gt; list of changes has been applied to that verification to user. @param sessionContext @param verificationId</p>	<p>This method will take in a verification ID and return the complete history of the &lt;br&gt; verification.&lt;br&gt; &lt;br&gt; For example, the banker might want to view the history of some verification he is &lt;br&gt; working on. On select of the verification (or verification ID), the system will return the VerificationHistoryDTO&lt;br&gt; The VerificationHistoryDTO stores the changes in status of verification like for eg: from PENDING to SUCCESSFUL.The original status is the status before updating the &lt;br&gt; verification. The final status is the status after updating the verification. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String verification ID, Verification id for which history data has to be fetched information about the user of the system. The information is used by the system for authentication purposes.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed.&lt;br&gt; @fcb.task OriginationTaskConstants.MNEM_OR_APPLICATION_VERIFICATION&lt;br&gt; @return VerificationHistoryResponse , The response class containing the history DTO's @throws FatalException&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationHistory	This method will take in a verification ID and return the complete history of the   verification.    For example, the banker might want to view the history of some verification he is   working on. On select the verification (or verification ID), the system will return the   list of changes has been applied to that verification to user. @param sessionContext @param verificationId	
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationHistory	This method will take in a verification ID and return the complete history of the   verification.    For example, the banker might want to view the history of some verification he is   working on. On select the verification (or verification ID), the system will return the   list of changes has been applied to that verification to user. @param sessionContext @param verificationId	This method calls the fetchVerificationHistory method in VerificationApplication service in Common services module. @param SessionContext, String @return VerificationHistoryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchReasonCodes	This method calls the fetchReasonCodes method in ReasonCodesApplication service in Common services module. @param SessionContext @param ReasonCodesDTO @return VerificationHistoryResponse	The method is used to fetch all the the offer documents that have been dispatched to the customer and are waiting to be accepted / executed. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @return OfferDocumentLinkageResponse @exception   {@link OriginationErrorConstants#CHECKLIST_NOT_FOUND} : FC_OR_OF_0043   This error occurs when:  The offer documents are retrieved for an application for which the offer document checklist has not been created
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchReasonCodes	This method calls the fetchReasonCodes method in ReasonCodesApplication service in Common services module. @param SessionContext @param ReasonCodesDTO @return VerificationHistoryResponse	The method is used to fetch the history of all the offer documents for the given application ID @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String offerDocReferenceId , Offer document reference id (Represents application id / submission id) @param String offerDocReferenceType , Offer document reference type (Represents whether application / submission) @return OfferDocumentLinkageHistoryResponse @exception   {@link OriginationErrorConstants#OFFER_HISTORY_NOT_FOUND} : FC_OR_OF_0048    {@link OriginationErrorConstants#OFFER_HISTORY_NOT_FOUND} : FC_OR_OF_0048   This error occurs when there are no history offer records for the given

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				application
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchReasonCodes	This method calls the fetchReasonCodes method in ReasonCodesApplication service in Common services module. @param SessionContext @param ReasonCodesDTO @return VerificationHistoryResponse	This method calls the fetchReasonCodes method in ReasonCodesApplication service in Common services module. @param SessionContext @param ReasonCodesDTO @return VerificationHistoryResponse
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchReasonCodes	This method calls the fetchReasonCodes method in ReasonCodesApplication service in Common services module. @param SessionContext @param ReasonCodesDTO @return VerificationHistoryResponse	This represents the fetchVerificationType method of the entity. @param SessionContext @param VerificationTypeDTO @return VerificationTypeInquiryResponse @throws FatalException
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	getWaivedVerifications	This method calls the getWaivedVerification method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId @return WaivedVerificationResponse	This method calls the fetchChecklistIndicator method in VerificationApplication service in Common services module. @param SessionContext, VerificationChecklistId @return VerificationIndicatorResponse
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationType	This represents the fetchVerificationType method of the entity. @param SessionContext @param VerificationTypeDTO @return VerificationTypeInquiryResponse @throws FatalException	This method will be called from UI when new verification task will be added in checklist. Internally  it will send an alert to applicant specifying to submit documents for verification if any pending.  The purpose of This method is only for sending an alert if any new verification task is later   added so that we can send alert specifying that documents needs to be verified. @param sessionContext @param verificationDTO @param applicationId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchVerificationType	This represents the fetchVerificationType method of the entity. @param SessionContext @param VerificationTypeDTO @return VerificationTypeInquiryResponse @throws FatalException	This represents the fetchVerificationType method of the entity. @param SessionContext @param VerificationTypeDTO @return VerificationTypeInquiryResponse @throws FatalException
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchBankCodeFields	The function would fetch valid bank code fields. @param sessionContext @param bankCode @return @throws FatalException	The function would fetch valid bank code fields. @param sessionContext @param bankCode @return @throws FatalException
IOriginationCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationCommonServicesAdapter	fetchBankCodeFields	The function would fetch valid bank code fields. @param sessionContext @param bankCode @return @throws FatalException	This method will return the name/description for given BankCode from Common Service. @param sessionContext @return
IAccountPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchVerifiedLinkedAccountsForParties	This method will execute query and fetch all verified linked accounts for the input party id's. @param partyIdArray @return @throws FatalException	This method will fetch all verified linked accounts for the bundle owners and validate that the external account that has been supplied as the new nominated account belongs to one of the bundle owners. @param partyIdArray @param accountId @return
IAccountPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchVerifiedLinkedAccountsForParties	This method will execute query and fetch all verified linked accounts for the input party id's. @param partyIdArray @return @throws FatalException	
IAccountPCAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchPartiesForLinkedAccount	This method will execute query and fetch all party id's for the input account Id. @param accountId @return @throws FatalException	This method validates the DDA account no specified in the settlement, checks whether the DDA account holder and the Loan Account holders are the same. @param componentPreferenceDTO @param partyToAccountRelationshipDTOs @param amendEvent @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchPartiesForLinkedAccount	This method will execute query and fetch all party id's for the input account Id . @param accountId @return @throws throws FatalException	Method to process Maintenance fees from External account @param applicationContext @param accountBasicDetailsDTO @param bundleFeeBatchData @param componentPreferenceDTO @return @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchPartiesForLinkedAccount	This method will execute query and fetch all party id's for the input account Id . @param accountId @return @throws throws FatalException	Method to process Maintenance fees from External account @param applicationContext @param accountBasicDetailsDTO @param bundleFeeBatchData @param componentPreferenceDTO @return @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchPartiesForLinkedAccount	This method will execute query and fetch all party id's for the input account Id . @param accountId @return @throws throws FatalException	
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchAllBeneficiaryAccountsForParty	Method to fetch all beneficiary accounts for party @param partyId @return CounterpartyDTO[] @throws FatalException	@throws FatalException @see com.ofss.fc.app.dda.service.core.IDemandDepositInquiryService#fetchBeneficiariesOfParty(com.ofss.fc.app.context.SessionContext, java.lang.String)
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fetchAllBeneficiaryAccountsForParty	Method to fetch all beneficiary accounts for party @param partyId @return CounterpartyDTO[] @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	getValueDateForCheque	This method is used to get upfront value date for a post dated cheque; Should be called by setting sessionContext.setServiceCallContextType(ServiceCallContextType.VALIDATE), so that cheque is not actually posted. @param applicationContext @param transactionCategoryId @param transactionDTO @return Date	This method obtains the value date for contribution payin request via cheque/external account from payments module during settlement payout instruction setup. @param sessionContext @param loanAccountId @param componentPreferenceDTO @param contributionAmount @return SettlementPayoutInquiryResponse @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	validateBeforeDeposit	This method should be called to deposit multiple cheques by giving the instrument details of the transaction. @param MI, InstrumentTransferCaptureDTO, instrumentTransferCaptureDTO <li> @link com.ofss.fc.app.pc.dto.input.InstrumentTransferCaptureDTO</li> <li> @return FundsExchangeResponse ,<li> @link com.ofss.fc.app.pc.dto.FundsExchangeResponse</li> @throws FatalException	This method will perform multiple cheque deposits along with cash deposit.  The MultiFundDepositApplication  <li>calls  {@link OutgoingClearingService#save} to validate details of each cheque before deposit.   </li><li>calls  {@link com.ofss.fc.app.dda.service.transaction.MultiFundDepositApplication#depositMultiFund(com.ofss.fc.app.dda.dto.transaction.MultiFundDepositDTO)} to deposit multiple cheques along with cash in the beneficiary account. This will return corresponding accounting amount tag details with credit leg information for each cheque and cash deposit processed in the transaction.   </li><li>calls  {@link com.ofss.fc.app.account.multifunddeposit.service.MultiFundDepositApplication#raiseAccounting(ApplicationContext, MultiFundDepositDTO, String, TransactionContext, FundsExchangeResponse, MultiFundDepositResponse)} to raise accounting for each cheque and cash deposit of multi fund deposit transaction. This will generate accounting amount tag details with debit leg information before raising the accounting.   </li><li>calls  {@link MultiFundDepositApplication#raiseAlert(MultiFundDepositDTO)} to send alert to the customer once Multi

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Fund deposit transaction is done successfully.&lt;/li&gt;  &lt;br /&gt; @fcb.param MI, ApplicationContext Indicates basic information about the interaction being initiated with the system. &lt;li&gt; {@link ApplicationContext}&lt;/li&gt; @fcb.param MI, MultiFundDepositDTO Indicates all the information related to multi fund  deposit(cashTransactionDTO,instrumentTransferCaptureDTO,wrappedPriceFeatureDTO) &lt;li&gt; {@link MultiFundDepositDTO}&lt;/li&gt; @return TransactionStatus &lt;li&gt;{@link com.ofss.fc.service.response.TransactionStatus}&lt;/li&gt; @throws FatalException</p>
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	fecthTransactionTemplate	<p>This method will fetch the template for the given template id and returns immediate credit threshold amount that is set in the template. @fcb.param MI, String templateId , It is the template id which is to be fetched. @return MoneyDTO, It is the immediate credit threshold amount that is set in the template. @throws FatalException</p>	<p>This method handle the cases where an immediate credit is required for a cheque deposit when the transaction amount is less than the immediate credit threshold amount set in the template. @param accountPCAdapter @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	fecthTransactionTemplate	This method will fetch the template for the given template id and returns immediate credit threshold amount that is set in the template. @fcb.param MI, String templateId , It is the template id which is to be fetched. @return MoneyDTO, It is the immediate credit threshold amount that is set in the template. @throws FatalException	This will generate the Multi Fund Deposit Data Transfer Object required for the DDA module to process fund transfer to the provided CASA account. @fcb.param MI, MultiFundDepositDTO, multiFundDepositDTO <li> {@link com.ofss.fc.app.account.dto.multifunddeposit.MultiFundDepositDTO}</li> @fcb.param MI, FundsExchangeResponse,fundsExchangeResponse <li> {@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @return MultiFundDepositDTO <li>{@link com.ofss.fc.app.dda.dto.transaction.MultiFundDepositDTO}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	liquidateMultiFundDepositTransaction	<p>This method will be called during multi fund deposit transaction for liquidation of bank draft .&lt;br&gt; This will generate the debit leg along with amount tag details required for processing the multi fund deposit transaction. This method will do the validations and processing for liquidation of bank draft. The accounting has to be raised separately as this will not process the accounting. @fcb.param MI, SessionContext, sessionContext</p> <p>&lt;li&gt;{@link com.ofss.fc.app.context.SessionContext}&lt;/li&gt; @fcb.param MI, BankDraftDTO, bankDraftDTO</p> <p>&lt;li&gt;{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}&lt;/li&gt; @fcb.param MI, LiquidationModeType, liquidationModeType &lt;li&gt; {@link com.ofss.fc.enumeration.pc.LiquidationModeType}&lt;/li&gt; @return BankDraftResponse &lt;li&gt;{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftResponse}&lt;/li&gt; @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	liquidateMultiFundDepositTransaction	<p>This method will be called during multi fund deposit transaction for liquidation of bank draft .&lt;br&gt; This will generate the debit leg along with amount tag details required for processing the multi fund deposit transaction. This method will do the validations and processing for liquidation of bank draft. The accounting has to be raised separately as this will not process the accounting. @fcb.param MI, SessionContext, sessionContext</p> <p>&lt;li&gt;{@link com.ofss.fc.app.context.SessionContext}&lt;/li&gt; @fcb.param MI, BankDraftDTO, bankDraftDTO</p> <p>&lt;li&gt;{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}&lt;/li&gt; @fcb.param MI, LiquidationModeType, liquidationModeType &lt;li&gt; {@link com.ofss.fc.enumeration.pc.LiquidationModeType}&lt;/li&gt; @return BankDraftResponse &lt;li&gt;{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftResponse}&lt;/li&gt; @throws FatalException</p>	<p>This method is called to liquidate the Bankers Cheque case in multi fund deposit transaction. @fcb.param MI, String, accountNo @fcb.param MI, InstrumentLogItemDTO , instrumentLogItem &lt;li&gt;{@link com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO}&lt;/li&gt; @return BankDraftResponse @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	liquidateMultiFundDepositTransaction	This method will be called during multi fund deposit transaction for liquidation of bank draft .  This will generate the debit leg along with amount tag details required for processing the multi fund deposit transaction. This method will do the validations and processing for liquidation of bank draft. The accounting has to be raised separately as this will not process the accounting. @fcb.param MI, SessionContext, sessionContext <li>{@link com.ofss.fc.app.context.SessionContext}</li> @fcb.param MI, BankDraftDTO, bankDraftDTO <li>{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}</li> @fcb.param MI, LiquidationModeType, liquidationModeType <li> {@link com.ofss.fc.enumeration.pc.LiquidationModeType}</li> @return BankDraftResponse <li>{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftResponse}</li> @throws FatalException	This will raise accounting in case of Bank Draft transaction during multi fund deposit transaction. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, FundDepositResponse, fundDepositResponse <li> {@link com.ofss.fc.app.dda.dto.transaction.FundDepositResponse}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	validateMultipleCheques	This method should be called to deposit multiple cheques by giving the instrument details of the transaction. @fcb.param MI, InstrumentTransferCaptureDTO, instrumentTransferCaptureDTO <li> {@link com.ofss.fc.app.pc.dto.input.InstrumentTransferCaptureDTO}</li> &@return FundsExchangeResponse ,<li>{@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @throws FatalException	
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	validateMultipleCheques	This method should be called to deposit multiple cheques by giving the instrument details of the transaction. @fcb.param MI, InstrumentTransferCaptureDTO, instrumentTransferCaptureDTO <li> {@link com.ofss.fc.app.pc.dto.input.InstrumentTransferCaptureDTO}</li> &@return FundsExchangeResponse ,<li>{@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @throws FatalException	This method validates all the cheque details of a multifund deposit transaction. This method fetches the error suggestions for each individual cheque of the transaction. This will be helpful to the user to know errors in individual cheque detail he has entered and allows him to quickly modify only those cheque details before performing the actual transaction. @fcb.param MI, MultiFundDepositDTO Indicates all the information related to multi fund deposit(cashTransactionDTO,instrumentTransferCaptureDTO,wrappedPriceFeatureDTO) <li> {@link MultiFundDepositDTO}</li> @return FundsExchangeResponse<li>{@link com.ofss.fc.app.pc.dto.FundsExchangeResponse}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	inquireChequeStatus	This method should be called to inquire the cheque status of all the records of the transaction for the given reference number. @fcb.param MI, String, refereneNumber @return ChequeStatusInquiryResponse ,<li>{@link com.ofss.fc.app.pc.dto.chequestatus.ChequeStatusInquiryResponse}</li> > @throws FatalException	This will fetch the Pseudo entries logged for the the given reference number and accountId. The pseudo entry is logged for the processing MFD where single statement.  Data to be fetched from Payments are the following: <ol> <li>Counterparty institution id value</li> <li>Instrument date</li> <li>Cheque number</li> <li>Counterparty account number</li> <li>Drawee name</li> <li>Instrument amount</li> </ol> @param sessionContext @param accountId @param referenceNumber @return @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	inquireBankDraft		This will fetch the Pseudo entries logged for the the given reference number and accountId. The pseudo entry is logged for the processing MFD where single statement.  Data to be fetched from Payments are the following: <ol> <li>Counterparty institution id value</li> <li>Instrument date</li> <li>Cheque number</li> <li>Counterparty account number</li> <li>Drawee name</li> <li>Instrument amount</li> </ol> @param sessionContext @param accountId @param referenceNumber @return @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	inquireBankDraft		
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountPCAdapter	doImmediateCreditByRef	@param sessionContext @param immediateCreditRequestDTO @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	doImmediateCreditByRef	@param sessionContext @param immediateCreditRequestDTO @return @throws FatalException	This method handle the cases where an immediate credit is required for a cheque deposit when the transaction amount is less than the immediate credit threshold amount set in the template. @param accountPCAdapter @throws FatalException
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	reverseMultiFundDepositTransaction	This method performs the reversal of the Multi fund deposit transaction on the same day. @param transactionReferenceNumber @throws FatalException	
IAccountPC Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountPCAdapter	reverseMultiFundDepositTransaction	This method performs the reversal of the Multi fund deposit transaction on the same day. @param transactionReferenceNumber @throws FatalException	@throws FatalException
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchMainFacilityForAccount	This method fetches the main facility linked with an accountId for the given value date @param accountId @param valueDate @return String @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchMainFacilityForAccount	This method fetches the main facility linked with an accountId for the given value date @param accountId @param valueDate @return String @throws FatalException	This method will be called by Adapter services of CASA/Loan to notify Asset Classification when new account is being opened.   Module to provide Account Id, Account Type, Product Code, Party Id(Primary), Facility Id. NPA module to update the main facility id if the module has given a sub-facility. The next classification date will be defaulted to current posting date. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String, accountId, , This field identifies the Loan or Overdraft(CASA) account for which asset classification is being performed. @fcb.param MI,String, accountType, , Identifies the product class to which the account belongs to. Valid accounts are CASA-Overdraft or LOAN @fcb.param MI,String, productCode, , Identifies the product code of the account. @fcb.param MI,String, partyId, , Identifies the party id of the account. @fcb.param OI,String, facilityId, , Identifies the facility id for the account. @fcb.param MI,String, ccy, , Identifies the account currency. @fcb.param MI,String, bu, , Identifies the business unit of the account. @fcb.param MI,String, accountBranchCode, , Identifies the branch code to which the account belongs. @param accountOpeningValueDate TODO @return {@link AccountClassificationResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchMainFacilityForAccount	This method fetches the main facility linked with an accountId for the given value date @param accountId @param valueDate @return String @throws FatalException	This method will be called by Adapter services of CASA/Loan to notify Asset Classification whenever the Party or the Facility linked to the account is changed. The next classification date will be defaulted to current posting date. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String, accountId, , This field identifies the Loan or Overdraft(CASA) account for which asset classification is being performed. @fcb.param MI,String, accountType, , Identifies the product class to which the account belongs to. Valid accounts are CASA-Overdraft or LOAN @fcb.param MI,String, productCode, , Identifies the product code of the account. @fcb.param MI,String, partyId, , Identifies the party id of the account. @fcb.param MI,String, facilityId, , Identifies the facility id for the account. @fcb.param MI,String, ccy, , Identifies the account currency. @fcb.param MI,String, bu, , Identifies the business unit of the account. @param effectiveDate TODO @return {@link AccountClassificationResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchMainFacilityForAccount	This method fetches the main facility linked with an accountId for the given value date @param accountId @param valueDate @return String @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchMainFacilityForAccount	This method fetches the main facility linked with an accountId for the given value date @param accountId @param valueDate @return String @throws FatalException	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchFacilityReviewLimitExpiryDays	This method fetches the days past facility review and limit expiry for the facility id @param facilityId @param accountId @return @throws FatalException	This method fetches values for Facility level rule related facts - Condition, covenant, insurance premium, limit expiry and facility review breach days @param mainFacilityId @return @throws FatalException
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchFacilityReviewLimitExpiryDays	This method fetches the days past facility review and limit expiry for the facility id @param facilityId @param accountId @return @throws FatalException	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchCovenantBreachDays	This method returns the number of breached days for a facility from the particular reference date. If the facility has multiple covenants, the result will be for earliest breached record. @param facilityId @return	This method fetches values for Facility level rule related facts - Condition, covenant, insurance premium, limit expiry and facility review breach days @param mainFacilityId @return @throws FatalException
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchConditionBreachDays	This method returns the number of breached days for a facility from the particular reference date. If the facility has multiple conditions, the result will be for earliest breached record. @param facilityId @return @throws FatalException	This method fetches values for Facility level rule related facts - Condition, covenant, insurance premium, limit expiry and facility review breach days @param mainFacilityId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchInsuranceBreachDays	This method returns the number of breached days for a facility from the particular reference date. If the facility has multiple insurance premium, the result will be for earliest breached record. @param facilityId @return	This method fetches values for Facility level rule related facts - Condition, covenant, insurance premium, limit expiry and facility review breach days @param mainFacilityId @return @throws FatalException
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchOverlineDaysForFacility	This method returns the days since the CASA account is in overline. The main facility linked to the account has to be passed to the service. @param mainFacilityId @return @throws FatalException	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchOverlineDaysForFacility	This method returns the days since the CASA account is in overline. The main facility linked to the account has to be passed to the service. @param mainFacilityId @return @throws FatalException	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchTODDAYSForAccount	This method returns the days since the CASA account is in OD. The account id is to be passed. @param accountId @return @throws FatalException	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchTODDAYSForAccount	This method returns the days since the CASA account is in OD. The account id is to be passed. @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	isAccountNominatedForFacilityFee	This method finds if the Account is nominated for debiting facility fee. An account nominated for facility debit is not to be Fully written-off. @param accountId @return	
IAssetClassificationLCMAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationLCMAAdapter	fetchApprovedLimit	This service fetches the approved limits for the facility @param facilityId @return	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return
IOriginationCreditDecisionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationCreditDecisionAdapter	sendSubmissionForDecision	This method is used to verify if the relationship selected in origination is the same role of applicant in ECIF @fcb.param MI, SessionContext,sessionContext, , Session Context which must contain the session context information such as user id, branch, branch code, channel etc.  @fcb.param MI,SubmissionDTO, submissionDTO, , Indicates the submission for which product information is to be set. @return SubmissionDecisionResponse, Indicates the submission decision response details as returned from the Credit Decision Engine @throws FatalException	This service is the interaction service to call the decision interface and receive the decision. Auto decision call is made to assess the application based on certain rules and policies configured in the decision engine. After creation of submission in the Lending Origination workflow, the details of entire submission are submitted to the credit decision interface which in turn will call credit decision engine for assessment. Details like interestSensitivityRate are fetched for each facility/product group and sent along with the submission details to the interface. A flag to indicate is the call is for auto-approval stage or auto-decision stage is also set and sent. The possible outcomes of the credit decision interface response are:- <ul><li>Approve</li> <li>Conditionally Approve</li> <li>Refer</li> <li>Refer Downsell</li> <li>Decline</li> <li>Continue</li> </ul> Depending on the outcome of the call to decision interface, further processing of submission will take place. <2.3>, from V2.3, if the outcome is Decline, Manual Credit Decision will be skipped and Applicaitons will be transited to state Application_Offer_Declined, and from which, client may initialize MCD again from app tracker or decline

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>task. @fcb.param MI,SessionContext,sessionContext,  ,Session Context which must contain the session  context information such as user id, branch, branch  code, channel etc @fcb.param MI,String,submissionId,  ,submissionId Unique Identifier for the submission  @return {@link  com.ofss.fc.app.origination.dto.lending.core.credit.decision.SubmissionDecisionResponse} The response  object of this call that contains the details of credit  decision of submission and enitites within  submission @exception&lt;br&gt; {@link  OriginationErrorConstants#SUBMISSION_NOT_FOU  ND} : FC_OR_DE_0055&lt;br&gt; This exception is  thrown if submission is not found for the given  submission id.&lt;br&gt; {@link  OriginationErrorConstants#SUBMISSION_HISTORY_  NOT_FOUND} : FC_OR_DE_0075&lt;br&gt; This  exception is thrown if submission history record is not  found for the submission id which is being edited or  amended.&lt;br&gt; {@link  OriginationErrorConstants#OR_INT_SENSITIVITY_R  ATE_NOT_MNT} : FC_OR_CD_0026&lt;br&gt; This  exception is thrown if interest sensitivity rate is not  maintained for the lending product group and  currency combination applied for in the  submission.&lt;br&gt; {@link  OriginationErrorConstants#OR_VALUATION_TYPE_  NULL} : FC_OR_CD_0029&lt;br&gt; This exception is  thrown for a collateral which requires valuation and if  the no valuation type is recommended by the credit  assessment engine.&lt;br&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response.   This also validates if the party is valid.Throws exception if the party is invalid or if it is not present   @return PartyInquiryResponse {@link com.ofss.fc.app.party.dto.core.PartyInquiryResponse} @throws FatalException	
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response.   This also validates if the party is valid.Throws exception if the party is invalid or if it is not present   @return PartyInquiryResponse {@link com.ofss.fc.app.party.dto.core.PartyInquiryResponse} @throws FatalException	Returns the basic details and demographic details form the part of response.   This also validates if the party is invalid or if it is not present   @return PartyInquiryResponse {@link com.ofss.fc.app.party.dto.core.PartyInquiryResponse} @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyBasicDetails	Returns the basic details and demographic details form the part of response.   This also validates if the party is valid.Throws exception if the party is invalid or if it is not present   @return PartyInquiryResponse {@link com.ofss.fc.app.party.dto.core.PartyInquiryResponse} @throws FatalException	This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyAddress	Function to fetch the partyAddress based on the partyAddressKeyDTO  The keys for the party address will be @param sessionContext @param partyAddressDTO @return PartyAddressResponse @throws FatalException	Function to fetch the partyAddress based on the partyAddressKeyDTO  The keys for the party address will be @param sessionContext @param partyAddressDTO @return PartyAddressResponse @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyWarningIndicator	This represents the fetchPartyWarningIndicator method of the entity. @return {@link com.ofss.fc.app.party.dto.core.PartyWarningIndicatorInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException	This represents the fetchPartyWarningIndicator method of the entity. @return {@link com.ofss.fc.app.party.dto.core.PartyWarningIndicatorInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the CCI Welcome Letter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the coverLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates CCIWelcomeLetterReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the CCIWelcomeLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the coverLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the coverLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates CoverReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the coverLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method will populate documaker request in case of Redispatch. OBPSUPPFIX23-603 Modified the method to merge the changes to release R2.3 @param sessContext @param domainObjectDTO @return DoPublishFromFactoryRequest @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the HypothecationLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the HypothecationLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates HypothecationReceivingParty attributes for the HypothecationLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the HypothecationLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the MortgageLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates PersonalGuaranteeReceivingParty attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the PersonalGuaranteeLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the MortgageLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Collateral attributes for the MortgageLetter extract. It performs following steps: <p> 1.Calls FacilityInquiryApplicationService.fetchFacilityInfo on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Calls CollateralApplicationService.fetchCollateralByBorrowingEntity <p> 4.Populates the Collateral related attributes @param SessionContext @param facilityId @return Collateral @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the MortgageLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates MortgageReceivingParty attributes for the MortgageLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the MortgageLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Collateral attributes for the offerLetter extract. It performs following steps: <p> 1.Calls ProposedFacilityInquiryApplicationService.fetchCollateralsForProposedFacility on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityResponse @param applicationId @param facilityId @return Collateral @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates the Product Group Applicant attributes for the offerLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicants attributes for the offerLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates OfferReceivingParty attributes for the offerLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method populates Applicant attributes of Receiving Party for the offerLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse	This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the CCI Welcome Letter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the coverLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates CCIWelcomeLetterReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the CCIWelcomeLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the coverLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the coverLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates CoverReceivingParty attributes for the coverLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the coverLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method will populate documaker request in case of Redispatch. OBPSUPPPFIX23-603 Modified the method to merge the changes to release R2.3 @param sessContext @param domainObjectDTO @return DoPublishFromFactoryRequest @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the HypothecationLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the HypothecationLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates HypothecationReceivingParty attributes for the HypothecationLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the HypothecationLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the MortgageLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Collateral attributes for the PersonalGuaranteeLetter extract. It performs following steps: <p> 1.Calls CollateralApplicationService.fetchCollateralDetails on basis of collateralId <p> 2.Fetches collateralResponse <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityId @param collateralId @return Collateral @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates PersonalGuaranteeReceivingParty attributes for the PersonalGuaranteeLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the PersonalGuaranteeLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the MortgageLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Collateral attributes for the MortgageLetter extract. It performs following steps: <p> 1.Calls FacilityInquiryApplicationService.fetchFacilityInfo on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Calls CollateralApplicationService.fetchCollateralByBorrowingEntity <p> 4.Populates the Collateral related attributes @param SessionContext @param facilityId @return Collateral @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the MortgageLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates MortgageReceivingParty attributes for the MortgageLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the MortgageLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	Function for inquiring the details of the party.  It calls the QueryPersonDetailsAdapter and performs the operation as required @param SessionContext @param partyId @param extPartyId @returns MDMPartyInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Collateral attributes for the offerLetter extract. It performs following steps: <p> 1.Calls ProposedFacilityInquiryApplicationService.fetchCollateralsForProposedFacility on basis of facilityId <p> 2.Fetches borrowingEntityId <p> 3.Populates the Collateral related attributes @param SessionContext @param facilityResponse @param applicationId @param facilityId @return Collateral @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates the Product Group Applicant attributes for the offerLetter extract. @param sessionContext @param loanProductApplicantGroupDTO @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicants attributes for the offerLetter extract. @param sessionContext @param loanProductGroupDto @param outboundDocumentRequestDTO @return Applicant @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates OfferReceivingParty attributes for the offerLetter extract. @param sessionContext @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method populates Applicant attributes of Receiving Party for the offerLetter extract. @param sessionContext @param applicantProductGroupReferenceDTOs @param outboundDocumentRequestDTO @return ReceivingParty @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDetails	This method will fetch party details using partyId only. @param sessionContext @param partyDTO @return PartyInquiryResponse @throws FatalException	This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchRoleBySpid	This returns the role id from SPID @param spid @return @throws FatalException	This populates the broker/introducer parent details @param iIntegrationPartyAdapter @param spid @return @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchRoleBySpid	This returns the role id from SPID @param spid @return @throws FatalException	@param spid @return ServiceProviderDTO @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchRoleBySpid	This returns the role id from SPID @param spid @return @throws FatalException	This returns the RoleDTO from the SPID @param spid @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchRoleBySpid	This returns the role id from SPID @param spid @return @throws FatalException	This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchCorporateDetails		This method will return corporate details @param organizationDTO @return PartyInquiryResponse
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchCorporateDetails		This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyFinancialProfile	Function for inquiring the financial details of a party. @param SessionContext @param partyFinancialProfileDTO @returns PartyFinancialProfileInquiryResponse	Function for inquiring the financial details of a party. @param SessionContext @param partyFinancialProfileDTO @returns PartyFinancialProfileInquiryResponse
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchPartyDeclaration	Function for the questions answers related to party @param SessionContext @param partyDeclarationDTO @returns PartyDeclarationInquiryResponse	Function for the questions answers related to party @param SessionContext @param partyDeclarationDTO @returns PartyDeclarationInquiryResponse



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchAllContactPointsByPartyDetails	@param partyId @return @throws FatalException	
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchAllContactPointsByPartyDetails	@param partyId @return @throws FatalException	This method is used to populate broker details @param lendingApplication @return AgentDetails @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchAgentArrangementLinkage	@param sessionContext @param agentArrangementLinkageDTO @return @throws FatalException	
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchLegalName	This return the legal name for the party @param sessionContext @param partyId @return @throws FatalException	This populates the AgentDetails for the submission @param sessionContext @param iIntegrationPartyAdapter @param submissionId @return @throws FatalException
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchLegalName	This return the legal name for the party @param sessionContext @param partyId @return @throws FatalException	This return the legal name for the party @param sessionContext @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchFinancialTemplate	<p>This represents the fetchFinancialTemplate method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,java.lang.String,templateId, It is a unique id that identifies the financial template. Using this id user can fetch the financial template corresponding to it @return {@link com.ofss.fc.app.party.dto.financial.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method populates the required financial details @param sessionContext @param partyresponse @param integrationPartyAdapter @param partyId @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchFinancialTemplate	<p>This represents the fetchFinancialTemplate method of the entity. @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param  MI,java.lang.String,templateId, It is a unique id that identifies the financial template. Using this id user can fetch the financial template corresponding to it @return {@link com.ofss.fc.app.party.dto.financial.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the fetchFinancialTemplate method of the entity. @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param  MI,FinancialTemplateDTO,financialTemplateDTO, ,financialTemplateDTO TODO: Explain the financialTemplateDTO above @return {@link com.ofss.fc.app.party.dto.finance.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\integration\documaker\IntegrationPartyAdapter	fetchFinancialTemplate	<p>This represents the fetchFinancialTemplate method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,java.lang.String,templateId, It is a unique id that identifies the financial template. Using this id user can fetch the financial template corresponding to it @return {@link com.ofss.fc.app.party.dto.financial.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the fetchFinancialTemplate method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,java.lang.String,templateId, It is a unique id that identifies the financial template. Using this id user can fetch the financial template corresponding to it @return {@link com.ofss.fc.app.party.dto.financial.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIntegrationPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\integration\documaker\IIntegrationPartyAdapter	fetchFinancialTemplateForPartyType	This represents the fetchFinancialTemplate method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FinancialTemplateDTO,financialTemplateDTO, ,financialTemplateDTO TODO: Explain the financialTemplateDTO above @return {@link com.ofss.fc.app.party.dto.finance.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the fetchFinancialTemplate method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,FinancialTemplateDTO,financialTemplateDTO, ,financialTemplateDTO TODO: Explain the financialTemplateDTO above @return {@link com.ofss.fc.app.party.dto.finance.template.FinancialTemplateInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	This operation gives the details of the CASA account balances such  as the available balance, the uncleared balance and the earmarked balance   using Balance Enquiry  These details are returned as a data transfer object of type DDABalanceDTO  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, , Business Context holds basic information about the transaction.  It contains the monetary value of the amount , it's currency and the currency conversion rate. etc  @return ATMBalanceResponse A

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>response object which holds the details of the&lt;br&gt; balances for the given account.The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS} &lt;/ul&gt; &lt;br&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D} &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_N T_FLG_CLS} &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED} &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#NO_OU TSTANDING_PO} &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_C LOSED_TODAY} &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_B AL_NT_ZERO} &lt;li&gt;{@link ErrorConstants#MID_ECH_INVALID_ACCT_NO} &lt;/ul&gt; &lt;li&gt;{@link ErrorConstants#MID_EUT_INVALID_USERID} &lt;li&gt;{@link ErrorConstants#MID_EST_INV_BANK} &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation should be use to inquire upon the statement of&lt;br&gt; accounts for a CASA account. The operation fetches all the transactions &lt;br&gt; done on a particular account in the specified period. This includes both&lt;br&gt; customer-initiated transactions and system initiated transactions like&lt;br&gt; interest and SC capitalization. &lt;br&gt; The statement of accounts for the current period fetches the details of all&lt;br&gt; transactions conducted during the current statement period. Similarly the&lt;br&gt; previous statement period fetches all the transactions performed during&lt;br&gt; the previous statement period.&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>payment transaction.&lt;br&gt; @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.&lt;br&gt; @return ATMStatementResponse The response contains details of the all transactions as well as&lt;br&gt; the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; following will be in the response&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_I NVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_BANK}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_C ORRDFN_NOT_FOUND}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_DUM MY_ERROR_CODE}&lt;/li&gt; &lt;/ul&gt; @throws FatalException See the return paragraph above</p>
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	This method calls the CASA fee-management service for levying channel usage fees @param context the session context to be used in the request @param accountID the account id against which fees management needs to be called @param eventId the event id against which fees management needs to be called @param transactionAmount the transaction amount against which fees management needs to be called @param messageType the message type against which fees management needs to be called @param transactionType the transaction type(debit/credit) against which fees management needs to be called @param transactionDate the transaction date against which fees management needs to be called @param acctCurrencyCode the account currency code against which fees management needs to be called
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	fetchFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	This method calls the CASA fee-management service for reversing the levied channel usage fees @param SessionContext the session context to be used in the request @param accountID the account id to be used in reversing the levied fees @param eventId the event id against which reversal of fees needs to be called @param messageType the message type to be used in reversal of fees @param transactionType the transaction type (credit/debit) to be used in reversal of fees
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	This operation gives the details of the CASA account balances such  as the available balance, the uncleared balance and the earmarked balance   using Balance Enquiry  These details are returned as a data transfer object of type DDABalanceDTO  @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context which must contain the ATM Key.  @fcb.param MI,ISOBusinessContext,data, ,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Business Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate. etc&lt;br&gt; @return ATMBalanceResponse A response object which holds the details of the&lt;br&gt; balances for the given account.The transaction status will indicate the success or failure of a transaction. The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS} </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LINK_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation enables you to perform a earmark removal against a POS Terminal GL&lt;br&gt; by passing the account to be debited and corresponding GL to be credited&lt;br&gt; as input. @fcb.param MI,POSChannelContext,channelContext, , POS Channel Context which must contain the POS Key.&lt;br&gt; @fcb.param MI,ISOBusinessContext,data, ,BusinessContext Context holds basic information about the transaction.&lt;br&gt; It contains the monetary value of the amount , it's currency and the currency conversion rate.&lt;br&gt; @return response An instance of {@link com.ofss.fc.app.channel.response.dto.POSBalanceResponse} containing the required information along with status.The transaction status will indicate the success or failure of a transaction.The following sections indicate the completion of the transactions with reasons&lt;br&gt; &lt;br&gt; &lt;b&gt;Success&lt;/b&gt; &lt;ul&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure due to warning&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OVR_EXCEPT_PRES_01}&lt;br&gt; Transaction requires authorisation for one of the reasons mentioned below &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#HOLD_FUND_ON_ACCT}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_AMB}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_</li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>FUNDS_OVL_HOLD_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_OVL_EXCEED}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_FUNDS_TOD_HOLD}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_LIN K_FAILED}&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; &lt;br&gt; &lt;b&gt;Transaction failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#OL_VOI D}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ResponseCodeType#CH_VOI D}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_SYN C_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#FC_FAT AL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERR OR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ERR_INS UFF_FUNDS}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INSUFF_ FUNDS_OVL}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OVER_I B_WDR_LIM}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#INV_AC CT_CODE}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#OLTP_I NT_ERROR}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_I NFO_CHANGED}&lt;/li&gt; &lt;li&gt;{@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>com.ofss.fc.enumeration.ReplyMessageType#RECORD_NOT_FOUND}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#INV_PSBK_BAL}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#DB_ERROR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_CLOSED_TODAY}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#ACCT_BLOCKED_ERR}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#NODRONACCT}&lt;/li&gt; &lt;li&gt; {@link com.ofss.fc.enumeration.ReplyMessageType#TOO_MANY_WITHDRAWALS}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @throws FatalException Please refer to the documentation above for &lt;i&gt;Transaction failure&lt;/i&gt; section</p>
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	<p>This operation should be use to inquire upon the statement of&lt;br&gt; accounts for a CASA account. The operation fetches all the transactions &lt;br&gt; done on a particular account in the specified period. This includes both&lt;br&gt; customer-initiated transactions and system initiated transactions like&lt;br&gt; interest and SC capitalization. &lt;br&gt; The statement of accounts for the current period fetches the details of all&lt;br&gt; transactions conducted during the current statement period. Similarly the&lt;br&gt; previous statement period fetches all the transactions performed during&lt;br&gt; the previous statement period.&lt;br&gt; @fcb.param MI,ATMChannelContext,channelContext, , ATM Channel Context will hold the CASA account number of the Customer which will get debited during bill</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>payment transaction.&lt;br&gt; @fcb.param MI,ISOBusinessContext, , Holds Additional information and can be validated by calling Rule.&lt;br&gt; @return ATMStatementResponse The response contains details of the all transactions as well as&lt;br&gt; the account details and its balances etc. along with {@link TransactionStatus} indicating success or failure&lt;br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; following will be in the response&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enum.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; Error Reason may be one of the following &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EUT_I NVALID_USERID}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_BANK}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EC_C ORRDFN_NOT_FOUND}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_EST_I NV_ACCT_NO}&lt;/li&gt; &lt;li&gt;{@link com.ofss.fc.common.FAPIErrorConstants#MID_DUM MY_ERROR_CODE}&lt;/li&gt; &lt;/ul&gt; @throws FatalException See the return paragraph above</p>
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPricingAdapter	reverseFeeForChannel	This method calls the casa fee-management service for levying channel usgae fees @param SessionContext	This method calls the CASA fee-management service for reversing the levied channel usage fees @param SessionContext the session context to be used in the request @param accountID the account id to be used in reversing the levied fees @param eventId the event id against which reversal of fees needs to be called @param messageType the message type to be used in reversal of fees @param transactionType the transaction type (credit/debit) to be used in reversal of fees
IDeferredWebServiceRequestPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IDeferredWebServiceRequestPollerAdapter	initiate	This method will initiate casa poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	This method will initiate webservice poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.account.IDeferredWebServiceRequestPollerAdapter#initiate(com.ofss.fc.app.context.SessionContext)
IDeferredWebServiceRequestPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IDeferredWebServiceRequestPollerAdapter	initiate	This method will initiate casa poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller , delayed transaction poller , FTP webservice poller. Please add more in this method Additionally we also have an extension for consulting pollers @param sessionContext
IDeferredWebServiceRequestPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IDeferredWebServiceRequestPollerAdapter	initiate	This method will initiate casa poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	Initialize all singletons used in the product . Currently supports , channel , businessobject, referenceobject, attribute definition , udf singletons. Please add more in this method @param sessionContext
ICollectionDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionDMSAdapter	generateDocument		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollectionDMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollectionDMSAdapter	generateAndDeliverDocument		This method generates and deliver document to Party. IOutboundDocumentApplicationService (for BIP/Documaker) is retrieved based on document type id. Using returned IOutboundDocumentApplicationService document is generated and delivered to respective party based on its configured contact preferences. Mandatory parameters are {@link OutboundDocumentRequestDTO} {@link DocumentDTO}[OwnerId,DocumentTypeId]. @param outboundDocumentRequestDTO - {@link OutboundDocumentRequestDTO} @return outboundDocumentResponse - {@link OutboundDocumentResponse} @throws Exception
IDocumentManifestFileAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentManifestFileAdapter	createManifestFile	This method is use to create the manifest file. @param sessionContext @param groupedElementOutboundDocReqDTOList @param numberOfInserts @param noOfcopiesString @param noOfInsertsString @return File @throws FatalException	Creates the manifest and zip attachments.If the document type attached to the inserts then, inserts also attach in zip file. @param sessionContext the session context @param groupedElementOutboundDocReqDTOList the grouped element out bound doc req dto list @return the document delivery response[] @throws FatalException the fatal exception @throws IOException Signals that an I/O exception has occurred.
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	taxationRefundSettlement	@param context @param instruction @param amountForSettlement @param treatmentType @throws FatalException	This method is used to refund the tax amount to the customer. This method will take the {@link AccountTaxRefundDetailsDRO} as input.  User has multiple options to choose to refund the amount. @param taxRefundDTO @param sessionContext



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	taxationRefundSettlement	@param context @param instruction @param amountForSettlement @param treatmentType @throws FatalException	
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	doBatchSettlement	@param taxationSettlementRequest @return @throws FatalException @throws FCRException	@param taxSettlementRequest @throws FCRException @throws FatalException
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	addComponentPreference		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	addComponentPreference		<p>This service is used during tax absorb account maintenance. This is required in the following situations. &lt;li&gt;To provide a provision to deduct Tax for the active/ closed Term Deposits from the specified CASA account/ GL account, when the customer is identified as a taxable customer.&lt;/li&gt;</p> <p>&lt;li&gt;Customer wants to pay the tax for the income earned from a CASA account instead of deducting the tax from the income amount.&lt;/li&gt;&lt;br&gt;This method is exposed to perform insert operation for absorb account. The following is the algorithm : &lt;br&gt; &lt;li&gt;Call {<a href="#">@link ITaxationSettlementApplicationAdapter#addComponentPreference</a>} by passing {<a href="#">@link ComponentPreferenceDTO</a>} to capture the settlement information in the settlement module.&lt;/li&gt; &lt;li&gt;Settlement module returns the reference number for the above maintenance&lt;/li&gt; &lt;li&gt;Read the reference number and set in the {<a href="#">@link DepositDetailsDTO</a>}</p> <p>&lt;li&gt;Prepare {<a href="#">@link DepositDetails</a>} by calling {<a href="#">@link TaxAssembler#makeDepositInformationObject</a>}</p> <p>&lt;li&gt;Then make a call {<a href="#">@link TaxRecoveryService#maintainAbsorbAccount(DepositDetails)</a>} by passing the {<a href="#">@link DepositDetails</a>}</p> <p>@param sessionContext @param depositInformationDTO</p>
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	updateComponentPreference		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	updateComponentPreference		<p>This service is used during tax absorb account maintenance. This is required in the following situations. &lt;li&gt;To provide a provision to deduct Tax for the active/ closed Term Deposits from the specified CASA account/ GL account, when the customer is identified as a taxable customer.&lt;/li&gt;</p> <p>&lt;li&gt;Customer wants to pay the tax for the income earned from a CASA account instead of deducting the tax from the income amount.&lt;/li&gt;&lt;br&gt;This method is exposed to perform insert operation for absorb account. The following is the algorithm : &lt;br&gt; &lt;li&gt;Call {@link ITaxationSettlementApplicationAdapter#addComponentPreference} by passing {@link ComponentPreferenceDTO} to capture the settlement information in the settlement module.&lt;/li&gt; &lt;li&gt;Settlement module returns the reference number for the above maintenance&lt;/li&gt; &lt;li&gt;Read the reference number and set in the {@link DepositDetailsDTO}&lt;/li&gt; &lt;li&gt;Prepare {@link DepositDetails} by calling {@link TaxAssembler#makeDepositInformationObject}&lt;/li&gt; &lt;li&gt;Then make a call {@link TaxRecoveryService#maintainAbsorbAccount(DepositDetails)} by passing the {@link DepositDetails}&lt;/li&gt; @param sessionContext @param depositInformationDTO</p>
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	fetchComponentPreference		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	fetchComponentPreference		
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	closeComponentPreference		
ITaxationSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\taxation\ITaxationSettlementApplicationAdapter	closeComponentPreference		<p>This service is used during tax absorb account maintenance. This is required in the following situations. &lt;li&gt;To provide a provision to deduct Tax for the active/ closed Term Deposits from the specified CASA account/ GL account, when the customer is identified as a taxable customer.&lt;/li&gt;</p> <p>&lt;li&gt;Customer wants to pay the tax for the income earned from a CASA account instead of deducting the tax from the income amount.&lt;/li&gt;&lt;br&gt;This service will take the following parameters</p> <p>&lt;li&gt;DepositInformationDTO&lt;/li&gt;&lt;br&gt; See {@link DepositDetailsDTO} for more information about this DTO.&lt;br&gt;This method is exposed to perform close operation for absorb account. @param depositInformationDTO</p>
ITDEventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\td\ITDEventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITDEventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	This method invokes executePreferenceOnline method of SettlementExecutionService for Batch Transactions. It reads the Settlement Response and updates the Transaction Detail @param sessionContext @param transactionDetail @param depositDTO @return @throws FatalException @throws FCRException
ITDEventExtensionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDEventExtensionAdapter	processExtension	This is a common method which can be used in all the TD related events to provide extension.	Post renewal of a Matured Deposit, This method is invoked to determine if any Blocks or Liens are present on the Matured Deposit. <p> If Lien is marked, then the Lien is removed from the Matured Deposit and carried forward onto the New Renewed Deposit. On the Matured Deposit the liened principal is subtracted from the Principal Balance. </p> <p> Similarly, if there is an Account or Deposit level Block, then the Blocks are removed from Matured Deposit and Placed on the New Renewed Deposit. </p> @param applicationContext @param deposit @param newDeposit @param account @throws FatalException
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForAccountUpdates	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_AccountFeederWrapperDTO @throws FatalException	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_AccountFeederWrapperDTO @throws FatalException
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForAccountUpdates	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_AccountFeederWrapperDTO @throws FatalException	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_WarningIndicatorFeedDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForAccountUpdates	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_AccountFeederWrapperDTO @throws FatalException	This method will be called from the respective services to get the updates and sent that update to the collection. @param p_AccountFeederWrapperDTO
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForAccountUpdates	This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_AccountFeederWrapperDTO @throws FatalException	
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForWarningIndicatorUpdates		This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param p_WarningIndicatorFeedDTO @throws FatalException
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForWarningIndicatorUpdates		This method will be called from the respective services to get the updates and sent that update to the collection. @param accountId @param List <WarningIndicatorAuditItem>
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForWarningIndicatorUpdates		This method will be called for removal of all the warning indicators from collection. @param accountId @param List <WarningIndicatorAuditItem>
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForWarningIndicatorUpdates		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForArrears		This method will be called from the respective services to get the updates and sent that update to the collection. @param accountId @param arrearList
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForArrears		
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForCustomerInitiatedBal		This method will be called from the DDAAccountCollectionIntimationService services to update the collection. @param feederPaymentsWrapperDTO @throws FatalException
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForCustomerInitiatedBal		This method will be called from the respective services to get the updates and sent that update to the collection. @param feederPaymentsWrapperDTO
IDDACollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDACollectionAdapter	updateCollectionForCustomerInitiatedBal		
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method returns calculated economic cost @param economicCostInquiryRequestDTO @return breakCostInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method will get the loan adapter and will call its calculateEconomicCost method to fetch the economic cost from ofsa. @param detailsDTO @return FundTransferPricingResponse
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method creates the OFSAA request,calls the web service and then uses the returned economic cost. @throws FatalException ,BusinessException
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method will get the loan adapter and will call its calculateEconomicCost method to fetch the economic cost from ofsa. @param detailsDTO @return FundTransferPricingResponse @throws FatalException
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method will be used to create request for OFSAA interaction.
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateEconomicCost		This method creates the OFSAA request,calls the web service and then uses the returned economic cost.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateQuoteExpiryDate		Method will be used to calculate quote expiry date. @param quoteGenerationDate @param accountId @return quoteExpiryDate
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateQuoteExpiryDate		Method will return quote expiry date for fixed period OD quote. @param accountId @param quotePeriod @param quoteGenerationDate @return
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateQuoteExpiryDate		This method will be used to store calculated economic cost details. @param ftpResponse
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	calculateQuoteExpiryDate		Method will be used to calculate quote expiry date. @param quoteGenerationDate @param accountId @return quoteExpiryDate
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	consumeOFSAAResponse		This method creates the OFSAA request,calls the web service and then uses the returned economic cost. @throws FatalException ,BusinessException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	consumeOFSAAResponse		This method will consume OFSAA request.It will simply create a response object and return. @param dto @return pricingResponse
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	consumeOFSAAResponse		This method will be called only from DemandDepositAccountService.chargeEconomicCostAndRaiseAccounting() to charge economic cost on fixed rate period OD account for online transaction as well as in batch processing. @param accountId @param preUtilizedAmount @param chargeableCreditAmount @param parentTransactionAmount @throws FatalException
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	consumeOFSAAResponse		This method will be used to create request for OFSAA interaction.
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	consumeOFSAAResponse		This method creates the OFSAA request,calls the web service and then uses the returned economic cost.
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	populateEconomicCostQuoteDTO		This method will be used for inquiring as well as maintaing economic cost details. If economic cost is already maintained and valid then it will return the details in response and if same is not maintained or not valid then it will be created first and then will be returned. @param sessionContext @param accountId @return economicCostQuoteResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOFSSAEconomicCostAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dataintegration\ofssa\IOFSSAEconomicCostAdapter	populateEconomicCostQuoteDTO		Method will be used to populate details in EconomicCostQuoteDTO related to fixed OD period. @param demandDepositAccount @param costQuoteDTO @return costQuoteDTO
IBBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBBranchBropDetailsAdapter	fetch		
IBBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBBranchBropDetailsAdapter	fetch		
IBBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBBranchBropDetailsAdapter	fetch		Fetches the batch operation details. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object containing the inquiry parameters. @return the batch operation inquiry response object <tt>BatchOperationInquiryResponse</tt>.
IBBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBBranchBropDetailsAdapter	fetch		Fetches the open batch operation details. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object containing the inquiry parameters. @return the batch operation inquiry response object <tt>BatchOperationInquiryResponse</tt>.
IBBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBBranchBropDetailsAdapter	fetch		This method is used to fetch values like posting date, next posting date and previous posting date which is required by the UI channels to construct sessionContext @param com.ofss.fc.app.context.SessionContext,sessionContext,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				,sessionContext @throws FatalException
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		The method will validate; 1 - If the batch for branch/teller for the given posting date is already opened/closed. 2 - If teller batch i) Is branch batch open for the posting date ii) Is any teller batch already open. 3 - If branch batch, is the batch open date either current date or next working date. 4 - Though the branch is open for two consecutive days, teller batches can be opened for only one posting date at any given time. The teller batch can be opened for the next posting date, only when it is closed for the Expected posting date.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		<p>&lt;p&gt; Checks if the cutoff is run for branch previous date. &lt;p&gt; The following logic is used. Fetch the record from flx_cs_branch_dates_b for given branch. &lt;li&gt;If FLG_CUTOFF_RUN_TODAY is Y, then the current date in flx_cs_branch_dates_b should be the branch previous date.&lt;/li&gt; &lt;li&gt;If FLG_CUTOFF_RUN_TODAY is N, then the current date in flx_cs_branch_dates_b should be the branch posting date.&lt;/li&gt;&lt;br&gt; &lt;p&gt; If not, then cutoff has not run for branch previous date, branch cannot be opened. @param batchOperation the &lt;tt&gt;BatchOperation&lt;/tt&gt; object.</p>
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		<p>&lt;p&gt; Returns whether the branch opening is allowed for the given date. &lt;p&gt; The following logic is used. If the given date is greater than the next working date of the Bank, thats aValidation error. If the cutoff has not completed for previous working date for the branch, thats a validation error. @param batchOperation the &lt;tt&gt;BatchOperation&lt;/tt&gt; object.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		<p> Returns the <tt>BranchDetailsInquiryResponse</tt> object for a given branch. @param branchCode the branch code. @return the <tt>BranchDetailsInquiryResponse</tt> object for a given branch. @throws FatalException
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		The method will validate if all teller batches are closed for given posting date.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		In case of split days, the branch batch which is opened for consecutive posting dates have to be closed in ascending order. For example, the branch batch with "10/18/2011" posting date should be closed first followed by the branch batch with "10/20/2011" posting date.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		This method will validate:  1 - If batch was opened for the posting date.  2 - If batch for the posting date is already closed.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		This method will check if any transactions are pending for the teller. The method will exclude any future dated currency indents.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetch		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetchBatchOperation		
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetchBatchOperation		Fetches the batch operation details. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object containing the inquiry parameters. @return the batch operation inquiry response object <tt>BatchOperationInquiryResponse</tt>.
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetchOpenBatches		
IBranchBropDetailsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchBropDetailsAdapter	fetchOpenBatches		This method is used to fetch values like posting date, next posting date and previous posting date which is required by the UI channels to construct sessionContext @param com.ofss.fc.app.context.SessionContext,sessionContext,sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<pre>This method is responsible for saving the account opening configuration done in Origination Systems. This method saves the following configuration in Loans: (1) Account Preference (2) Offset Account configuration (3) Account Payment Instruction (4) Repayment Schedule Configuration (5) Accelerated Payment Instruction (6) Rate Lock configuration Before saving the details, the following steps are first executed: (1) Validate account configuration w.r.t Product setup (2) Validate account configuration w.r.t Offer level parameters (3) Validate account configuration w.r.t Facility level setup On successful completion of the transaction, account configuration will be saved along with pricing recommendation. </pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , Session Context which must contain the session ticket and user id.  @return {@link SaveLoanAccountConfigurationResponse}. @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param	This represents the addMultipleLoanContributionPayinInstruction method of the entity. This method will be used to add multiple contribution instructions at a time. <ul> <li>sessionContext : contains the session related information.  <li>loanContributionPayinInstructionDTO : contains the details of the contribution  payin instruction.  </ul> @param sessionContext @param loanContributionPayinInstructionDTOArray @return transactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			eventId @param allowedSplitTypes @param productCode @return	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method is responsible for defining Periodic Credit Instruction for a Loan Account. @param sessionContext @param loanPeriodicCreditInstructionDTO @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method is responsible for defining Periodic Credit Instruction for a Loan Account when the account is not yet opened and party id is not yet stamped to it(specifically for Counter party settlement instruction). @param sessionContext @param loanPeriodicCreditInstructionDTO @param partyId @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method takes the LoanAccountSplitInstruction as input and processes it and Do the following: <li>PPF on Existing Account</li> <li>Open New Account Account</li> <li>Auto Disburse New Account</li> @param applicationContext @param splitInstruction @param transactionType @return @throws FatalException @throws OverridableException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<pre>This method is responsible for adding Loan Account Amend Configuration for the input account ID. Before the save operation, the following steps are executed: (1) Validate system constraints and throws appropriate exception. (3) Populate the Repayment Stage DTO Array from the input configurations (2) Validate if Amend transaction is allowed </pre> @param MI, SessionContext, sessionContext, , SessionContext which must contain the session ticket and user id.  @fcb.param MI, LoanAccountAmendConfigurationDTO, loanAccountAmendConfigurationDTO, ,This class represents the DTO for the LoanAccountAmendConfiguration entity.   @fcb.param MI, LoanAmendConfigurationParameterDTO, loanAmendConfigurationParameterDTO, ,This DTO holds parameters related to amend Account configuration   @return {@link LoanAccountAmendConfigurationInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction failure</i>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				section
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<pre>This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in Loans module. This method will be called in the context of an Instruction setup in Loans where in there is going be periodic transfer of funds from a Loan Account to transaction accounts or vice versa </pre> @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	<pre>This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in Loans module. This method will be called in the context of an Instruction setup in Loans where in there is going be periodic transfer of funds from a Loan Account to transaction accounts or vice versa </pre> @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method will be used for adding multiple disbursement instructions. The input  is an array of instructions (LoanDisbursementInstructionDTO) and session context.  It will check the system constraints and validates the business policy for each  disbursement instruction request.  @param sessionContext @param loanDisbursementInstructionDTOs @return TransactionStatus @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method does the update and delete of the disbursement instructions together.  The input is an array of instructions (LoanDisbursementInstructionDTO) and session  context. There is a maintenanceType in the input DTO which decides whether an  instruction is updated, deleted or added. Based on this maintenanceTyoe, the  disbursement instruction will be updated, deleted or added for that loan account.  Also, it will check the system constraints and the validate the business policy  for each disbursement instruction request.  @param sessionContext @param loanDisbursementInstructionDTOs @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method is used to create account payment instructions for a loan account  It does the following,    <ol> <li>Validate the inputs using system constraints, i.e.LoanAccountPaymentInstructionSystemConstraint.validateConstraints().</li> <li>Validate the Business policies, i.e.LoanAccountPaymentInstructionBusinessPolicy.validate().</li> <li>Invokes ComponentPreferenceApplicationService.addComponentPreference method via LoanSettlementApplicationAdapter.addSettlementDetails to obtain the settlementReferenceNumber.</li> <li>Invoke LoanProductService.fetchLoanProductPreferences method to fetch the loan product preferences associated with the customer account.</li> <li>Convert LoanAccountPaymentInstructionDTO to LoanAccountPaymentInstruction via LoanAccountPaymentInstructionAssembler.toDomainObject method</li> <li>Set LoanAccountPaymentInstruction.previousExecutionDate=null LoanAccountPaymentInstruction.scheduledExecutionDate=LoanAccountState.nextDueDate(Determine the nextInstallmentDueDate by loading LoanAccountState entity via LoanAccountStateRepository.getAccountState() method) //Determine the actualExecutionDate, based on whether the next installment due date falls on a working/non-working(holiday) day if(BusinessDateUtils.isWorkingDay(nextInstallmentDueDate)) { LoanAccountPaymentInstruction.actualExecutionDate=nextInstallmentDueDate } else { //Set

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>actualExecutionDate based on  HolidayProcessingMethodType  if(LoanProductPreferencesDTO.HolidayProcessingMethodType== PREVIOUS_WORKING_DAY)  LoanAccountPaymentInstruction.actualExecutionDate=BusinessDateUtils  .getPreviousWorkingDay(nextInstallmentDueDate)  else LoanAccountPaymentInstruction  .actualExecutionDate=BusinessDateUtils.getNextWorkingDay (nextInstallmentDueDate) }  LoanAccountPaymentInstruction.drawdownBehaviour=LoanProductPreferencesDTO .doDrawdownIn  (Determine the drawdown behaviour, EOD/BOD from LoanProductPreferencesDTO)  LoanAccountPaymentInstruction.holidayProcessingType=HolidayProcessingType  .PROCESS_ON_NEXT_WORKING_DAY or  HolidayProcessingType.PROCESS_ON_PREV_WORKING_DAY(Based on  LoanProductPreferencesDTO.drawdownOnNextWorkingDay=true or false)&lt;/li&gt; &lt;li&gt;Create account  payment instructions for the loan account i.e. using  LoanAccountPaymentInstructionService.createLoanAccountPaymentInstruction method passing the  LoanAccountPaymentInstruction&lt;/li&gt; &lt;/ol&gt;  @param sessionContext @param  loanAccountPaymentInstructionDTO @param  componentPreferenceDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	This method is used to create account payment instructions with counter party for a loan account  It does the following,    <ol> <li>Validate the inputs using system constraints, i.e.LoanAccountPaymentInstructionSystemConstraint.validateConstraints().</li> <li>Validate the Business policies, i.e.LoanAccountPaymentInstructionBusinessPolicy.validate().</li> <li>Invokes ComponentPreferenceApplicationService.addComponentPreference method via LoanSettlementApplicationAdapter.addSettlementDetails to obtain the settlementReferenceNumber.</li> <li>Invoke LoanProductService.fetchLoanProductPreferences method to fetch the loan product preferences associated with the customer account.</li> <li>Convert LoanAccountPaymentInstructionDTO to LoanAccountPaymentInstruction via LoanAccountPaymentInstructionAssembler.toDomainObject method</li> <li>Set LoanAccountPaymentInstruction.previousExecutionDate=null LoanAccountPaymentInstruction.scheduledExecutionDate=LoanAccountState.nextDueDate(Determine the nextInstallmentDueDate by loading LoanAccountState entity via LoanAccountStateRepository.getAccountState() method) //Determine the actualExecutionDate, based on whether the next installment due date falls on a working/non-working(holiday) day if(BusinessDateUtils.isWorkingDay(nextInstallmentDueDate)) { LoanAccountPaymentInstruction.actualExecutionDate=nextInstallmentDueDate } else { //Set

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>actualExecutionDate based on  HolidayProcessingMethodType  if(LoanProductPreferencesDTO.HolidayProcessingMethodType== PREVIOUS_WORKING_DAY)  LoanAccountPaymentInstruction.actualExecutionDate=BusinessDateUtils  .getPreviousWorkingDay(nextInstallmentDueDate)  else LoanAccountPaymentInstruction  .actualExecutionDate=BusinessDateUtils.getNextWorkingDay (nextInstallmentDueDate) }  LoanAccountPaymentInstruction.drawdownBehaviour=LoanProductPreferencesDTO .doDrawdownIn  (Determine the drawdown behaviour, EOD/BOD from LoanProductPreferencesDTO)  LoanAccountPaymentInstruction.holidayProcessingType=HolidayProcessingType  .PROCESS_ON_NEXT_WORKING_DAY or  HolidayProcessingType.PROCESS_ON_PREV_WORKING_DAY(Based on  LoanProductPreferencesDTO.drawdownOnNextWorkingDay=true or false)&lt;/li&gt; &lt;li&gt;Create account  payment instructions for the loan account i.e. using  LoanAccountPaymentInstructionService.createLoanAccountPaymentInstruction method passing the  LoanAccountPaymentInstruction&lt;/li&gt; &lt;/ol&gt;  @param sessionContext @param  loanAccountPaymentInstructionDTO @param  componentPreferenceDTO @param partyId</p>
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement	This method is used to add account payment instruction configuration as a part of structure solution process in account opening through origination. The instruction is set in the temporary configuration table till the account is not opened and once account opening is completed the instruction in the configuration table is moved to the actual instructions

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return</p>	<p>table.&lt;br /&gt; &lt;br&gt; &lt;b&gt;Algorithm :&lt;/b&gt; &lt;br /&gt; &lt;ol&gt;  &lt;li&gt;Validate the {@link LoanAccountPaymentInstructionDTO} and {@link ComponentPreferenceDTO} coming in as input for any kind of system constraints. &lt;/li&gt;Fetch the configuration instruction count by calling {@link LoanAccountPaymentInstructionConfigurationService#fetchConfigurationInstructionCount(String)} for the given arrangement id. &lt;/li&gt;Validate the {@link LoanAccountPaymentInstructionBusinessPolicy}. &lt;/li&gt;{@link ComponentPreferenceDTO} coming in as input is added to the Settlement details by calling {@link LoanSettlementApplicationAdapter#addSettlementDetails} which returns a settlement reference number which is in turn set in the {@link LoanAccountPaymentInstructionConfiguration}. &lt;/li&gt;Fetch loan product by calling {@link LoanProductService} which returns {@link LoanProductInquiryResponse} &lt;/li&gt;Get the drawdown behaviour from the in the {@link LoanProductInquiryResponse} and set it in {@link LoanAccountPaymentInstructionConfiguration}. &lt;/li&gt;Set the {@link HolidayProcessingTypes#PROCESS_ON_PREV_WORKING_DAY} and persists the instruction configuration response.&lt;/li&gt;&lt;br&gt; @param sessionContext @param loanAccountConfigurationDTO @param componentPreferenceDTO @param productCode @param offerId @return TransactionStatus @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @return	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetailsWithCounterParty	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @param offerId @param partyId @return	<pre>This method is responsible for saving the account opening configuration done in Origination Systems. This method saves the following configuration in Loans: (1) Account Preference (2) Offset Account configuration (3) Account Payment Instruction (4) Repayment Schedule Configuration (5) Accelerated Payment Instruction (6) Rate Lock configuration Before saving the details, the following steps are first executed: (1) Validate account configuration w.r.t Product setup (2) Validate account configuration w.r.t Offer level parameters (3) Validate account configuration w.r.t Facility level setup On successful completion of the transaction, account configuration will be saved along with pricing recommendation. </pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , Session Context which must contain the session ticket and user id.  @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				SaveLoanAccountConfigurationResponse}. @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetailsWithCounterParty	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @param offerId @param partyId @return	This method is responsible for defining Periodic Credit Instruction for a Loan Account when the account is not yet opened and party id is not yet stamped to it(specifically for Counter party settlement instruction). @param sessionContext @param loanPeriodicCreditInstructionDTO @param partyId @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetailsWithCounterParty	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @param offerId @param partyId @return	<pre>This method is responsible for adding Loan Account Amend Configuration for the input account ID. Before the save operation, the following steps are executed: (1) Validate system constraints and throws appropriate exception. (3) Populate the Repayment Stage DTO Array from the input configurations (2) Validate if Amend transaction is allowed </pre> @param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountAmendConfigurationDTO, loanAccountAmendConfigurationDTO, ,This class represents the DTO for the LoanAccountAmendConfiguration entity.   @fcb.param MI, LoanAmendConfigurationParameterDTO, loanAmendConfigurationParameterDTO, ,This DTO holds parameters related to amend Account configuration   @return {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				LoanAccountAmendConfigurationInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction failure</i> section
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addSettlementDetailsWithCounterParty	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode @param offerId @param partyId @return	This method is used to create account payment instructions with counter party for a loan account  It does the following,    <ol> <li>Validate the inputs using system constraints, i.e.LoanAccountPaymentInstructionSystemConstraint.validateConstraints().</li> <li>Validate the Business policies, i.e.LoanAccountPaymentInstructionBusinessPolicy.validate().</li> <li>Invokes ComponentPreferenceApplicationService.addComponentPreference method via LoanSettlementApplicationAdapter.addSettlementDetails to obtain the settlementReferenceNumber.</li> <li>Invoke LoanProductService.fetchLoanProductPreferences method to fetch the loan product preferences associated with the customer account.</li> <li>Convert LoanAccountPaymentInstructionDTO to LoanAccountPaymentInstruction via LoanAccountPaymentInstructionAssembler.toDomainObject method</li> <li>Set LoanAccountPaymentInstruction.previousExecutionDate=null LoanAccountPaymentInstruction.scheduledExecutionDate=LoanAccountState.nextDueDate(Determine the nextInstallmentDueDate by loading LoanAccountState entity via LoanAccountStateRepository.getAccountState() method) //Determine the actualExecutionDate, based on whether the next installment due date falls on a

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>working/non-working(holiday) day  if(BusinessDateUtils.isWorkingDay(nextInstallmentDueDate)) {  LoanAccountPaymentInstruction.actualExecutionDate=nextInstallmentDueDate } else { //Set actualExecutionDate based on HolidayProcessingMethodType  if(LoanProductPreferencesDTO.HolidayProcessingMethodType== PREVIOUS_WORKING_DAY)  LoanAccountPaymentInstruction.actualExecutionDate=BusinessDateUtils.getPreviousWorkingDay(nextInstallmentDueDate)  else LoanAccountPaymentInstruction.actualExecutionDate=BusinessDateUtils.getNextWorkingDay (nextInstallmentDueDate) }  LoanAccountPaymentInstruction.drawdownBehaviour=LoanProductPreferencesDTO .doDrawdownIn (Determine the drawdown behaviour, EOD/BOD from LoanProductPreferencesDTO)  LoanAccountPaymentInstruction.holidayProcessingType=HolidayProcessingType.PROCESS_ON_NEXT_WORKING_DAY or HolidayProcessingType.PROCESS_ON_PREV_WORKING_DAY(Based on LoanProductPreferencesDTO.drawdownOnNextWorkingDay=true or false)&lt;/li&gt; &lt;li&gt;Create account payment instructions for the loan account i.e. using LoanAccountPaymentInstructionService.createLoanAccountPaymentInstruction method passing the LoanAccountPaymentInstruction&lt;/li&gt; &lt;/ol&gt;  @param sessionContext @param loanAccountPaymentInstructionDTO @param componentPreferenceDTO @param partyId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method is responsible for updating Periodic Credit Instruction for a Loan Account. @param sessionContext @param loanPeriodicCreditInstructionDTO @return @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	<pre>This method is responsible for adding Loan Account Amend Configuration for the input account ID. Before the save operation, the following steps are executed: (1) Validate system constraints and throws appropriate exception. (3) Populate the Repayment Stage DTO Array from the input configurations (2) Validate if Amend transaction is allowed </pre> @param MI, SessionContext, sessionContext, , SessionContext which must contain the session ticket and user id.  @fcb.param MI, LoanAccountAmendConfigurationDTO, loanAccountAmendConfigurationDTO, ,This class represents the DTO for the LoanAccountAmendConfiguration entity.   @fcb.param MI, LoanAmendConfigurationParameterDTO, loanAmendConfigurationParameterDTO, ,This DTO holds parameters related to amend Account configuration   @return {@link LoanAccountAmendConfigurationInquiryResponse} @throws FatalException Please refer to the documentation above for <i>Transaction failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method does the update and delete of the disbursement instructions together.  The input is an array of instructions (LoanDisbursementInstructionDTO) and session  context. There is a maintenanceType in the input DTO which decides whether an  instruction is updated, deleted or added. Based on this maintenanceType, the  disbursement instruction will be updated, deleted or added for that loan account.  Also, it will check the system constraints and the validate the business policy  for each disbursement instruction request.  @param sessionContext @param loanDisbursementInstructionDTOs @return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	This method is used to update account payment instructions for a loan account  It does the following,    <ol> <li>Validate the inputs using system constraints, i.e.LoanAccountPaymentInstructionSystemConstraint.validateConstraints().</li> <li>Validate the Business policies, i.e. LoanAccountPaymentInstructionBusinessPolicy.validate().</li> <li>Invokes ComponentPreferenceApplicationService.updateComponentPreference method via LoanSettlementApplicationAdapter.updateSettlementDetails.</li> <li>Update account payment instruction for the loan account; i.e. using LoanAccountPaymentInstructionService.updateLoanAccountPaymentInstruction method.</li> </ol> @param sessionContext @param loanAccountPaymentInstructionDTO @param componentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	updateComponentPreference	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes @param productCode	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This represents the fetchLoanContributionPayinInstruction method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,LoanContributionPayinInstructionDTO,loanContributionPayinInstructionDTO,,loanContributionPayinInstructionDTO TODO: Explain the loanContributionPayinInstructionDTO above @return {@link com.ofss.fc.app.loan.dto.contribution.payin.instruction.LoanContributionPayinInstructionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method is responsible for fetching Periodic Credit Instruction defined for a Loan Account. @param sessionContext @param loanPeriodicCreditInstructionDTO @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method takes the LoanAccountSplitInstruction as input and processes it and Do the following: <li>PPF on Existing Account</li> <li>Open New Account Account</li> <li>Auto Disburse New Account</li> @param applicationContext @param splitInstruction @param transactionType @return @throws FatalException @throws OverridableException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method will provide all the active disbursement instructions for a loan account.  The disbursement instructions are fetched via a query from the repository, which Returns a list of the disbursement instructions for that account and then this list  is set in the response.  @param sessionContext @param loanDisbursementInstructionDTO @return LoanDisbursementInstructionInquiryResponse @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method will provide all the active disbursement instructions for a loan account.  The disbursement instructions are fetched via a query from the repository, which Returns a list of the disbursement instructions for that account and then this list  is set in the response.  @param sessionContext @param loanDisbursementInstructionDTO @return LoanDisbursementInstructionInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This private method will process a single contribution instruction request on theactionDate for the loan account.  Processing : <ul> <li>First, it will process the componentPreferences(i.e. settlement execution) associated with the contribution instruction.</li> <li>In case of successful execution of settlement, it will update the contributionAmount present in  LoanAccount via {@link LoanAccount.updateContributionAmount()} and LoanAccountBalances via  {@link LoanBalanceService.updateContributionReceived()}.</li> <li>raise accounting events.</li> <li>raise alerts</li> </ul> @param applicationContext @param loanContributionPayinInstruction @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	@return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method does business policy checks which are outside the loan domain. @param disbursementData @throws FatalException @return DisbursementValidationErrorType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method performs Disbursement posting considering the calculated fees.  It performs following tasks:-   <ul> <li>Inquire Settlement Details</li> <li>Prepare Fee Collection Details</li> <li>Post the Disbursement using Disbursement Posting Application</li> </ul> @param applicationContext @param deductionModeType @param eventId @param accountId @param disbursementType @param disbursementCount @param loanAutoDisbursementData @param wrappedPriceFeatureDTO @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	<p>This method is responsible to fetch loan account linkage details.   It checks whether Loan Account has been linked to other accounts for following purposes:-   <ul> <li><u>Offset Account:-</u></br> If flag is true then checks for details of linked offset account.</br></li> <li><u>Account Payment Instruction Facility Flag (Direct Debit):-</u></br> If flag is true then checks for details of Account Payment Instruction details</br></li> </ul> </p> @param sessionContext @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method is used to inquire account payment instructions for a given loan account. The instructions are fetched for the following purposes: <li>Closing the instructions on account closure via Early Final Settlement. <li>Fetching account linkage details which contains the instruction details as well. <b>Algorithm</b>  <ol> <li>Fetch LoanAccountPaymentInstruction using {@link LoanAccountPaymentInstructionService#fetchLoanAccountPaymentInstruction}</li> <li>Fetch ComponentPreferenceDTO by calling {@link LoanSettlementApplicationAdapter#inquireComponentPreferenceDetails} using settlement reference number in the payment instructions and the given account id.</li> <li>Return LoanAccountPaymentInstructionInquiryResponse after setting the payment instructions and the component preference DTO in it.</li> </ol> @param sessionContext @param loanAccountPaymentInstructionDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	This method is used to fetch the loan account payment instruction configuration which is created during the account opening through structure solution process in origination. After the account is opened, then the instruction is moved from the configurations to the actual instructions table.   <b>Algorithm :</b> <ol> <li>Fetches configuration details by calling {@link LoanAccountPaymentInstructionConfigurationService #fetchConfigurationInstructionCount(String)} which takes the arrangement Id as input.  <li>Set the {@link LoanAccountPaymentInstructionConfiguration} in the {@link LoanAccountPaymentInstructionDTO}.<li>Fetch ComponentPreferenceDTO by calling {@link LoanSettlementApplicationAdapter#inquireComponentPreferenceDetails} using settlement reference number in the payment instructions and the given account id.  </li> <li>Set the ComponentPreferenceDTO and LoanAccountPaymentInstructionDTO in the {@link LoanAccountPaymentInstructionInquiryResponse}.</li> @param sessionContext @param loanAccountPaymentInstructionDTO @return LoanAccountPaymentInstructionInquiryResponse @throws com.ofss.fc.infra.exception.FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	inquireComponentPreferenceDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	<pre> This method is responsible for saving the account opening configuration done in Origination Systems. This method saves the following configuration in Loans: (1) Account Preference (2) Offset Account configuration (3) Account Payment Instruction (4) Repayment Schedule Configuration (5) Accelerated Payment Instruction (6) Rate Lock configuration Before saving the details, the following steps are first executed: (1) Validate account configuration w.r.t Product setup (2) Validate account configuration w.r.t Offer level parameters (3) Validate account configuration w.r.t Facility level setup On successful completion of the transaction, account configuration will be saved along with pricing recommendation. </pre> @fcb.param MI, SessionContext, sessionContext, , Session Context which must contain the session ticket and user id.  @fcb.param MI, LoanAccountConfigurationDTO, loanAccountConfigurationDTO, , Session Context which must contain the session ticket and user id.  @return {@link SaveLoanAccountConfigurationResponse}. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is used to delete account payment instruction for a loan account  It does the following,    <ol> <li>Invokes ComponentPreferenceApplicationService.deleteComponentPreference method via LoanSettlementApplicationAdapter.deleteSettlementDetails. to delete the preference.  </li><li>Delete account payment instruction for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionService.deleteLoanAccountPaymentInstruction method.</li><li>Delete account payment instruction retry for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionRetryService.deleteLoanAccountPaymentInstructionRetry method.</li> </ol> @param sessionContext @param loanAccountPaymentInstructionDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This represents updateLoanAccountPaymentInstructionConfiguration method of the entity. It does the following,    <ol> <li>Check if instruction configuration corresponding to the given arrangement id is present. 1) If present,Delete the existing instruction configuration details  and add new instruction configuration details. 2) If instruction is not present then add the new instruction configuration details.</li> @param sessionContext @param loanAccountConfigurationDTO @param componentPreferenceDTO @param productCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This represents deleteLoanAccountPaymentInstructionConfiguration method of the entity. It does the following,    <ol> <li>Check if instruction configuration corresponding to the given arrangement id is present. 1) If present,Delete the existing instruction configuration details  and add new instruction configuration details.</li> @param sessionContext @param loanAccountConfigurationDTO @param componentPreferenceDTO @param productCode @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	deleteComponentPreferenceDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is responsible for reopening Periodic Credit Instruction for a Loan Account. @param sessionContext @param loanPeriodicCreditInstructionDTO @return @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is used to reopen account payment instructions for a loan account  It does the following,    <ol> <li>Invokes ComponentPreferenceApplicationService.reopenComponentPreference method via LoanSettlementApplicationAdapter.reopenSettlementDetails. to reopen the preference.  </li><li>Reopen account payment instruction for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionService.reopenLoanAccountPaymentInstruction method.</li><li>Reopen account payment instruction retry for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionRetryService.reopenLoanAccountPaymentInstructionRetry method.</li> </ol> @param sessionContext @param loanAccountPaymentInstructionDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reopenSettlementDetails	This method reopen the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is responsible for closing Periodic Credit Instruction for a Loan Account. @param sessionContext @param loanPeriodicCreditInstructionDTO @return @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	closeComponentPreference	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is used to close account payment instructions for a loan account  It does the following,    <ol> <li>Invokes ComponentPreferenceApplicationService.closeComponentPreference method via LoanSettlementApplicationAdapter.closeSettlementDetails. to delete the preference.  </li> <li>Close account payment instruction for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionService.closeLoanAccountPaymentInstruction method.</li> <li>Close account payment instruction retry for the loan account&nbsp; i.e. using LoanAccountPaymentInstructionRetryService.closeLoanAccountPaymentInstructionRetry method.</li> </ol> @param sessionContext @param loanAccountPaymentInstructionDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchByVersion	This method fetches the ComponentPreferenceDTO stored against the reference number and the accountId by its version. @param sessionContext @param componentPreferenceDTO @return	This method fetch the ComponentPreferenceDTO by its version. @param componentPreferenceDto
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchByVersion	This method fetches the ComponentPreferenceDTO stored against the reference number and the accountId by its version. @param sessionContext @param componentPreferenceDTO @return	This represents the fetchByVersion method of the entity. @param sessionContext @param loanDisbursementInstructionDTO @return DomainObjectDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchByVersion	This method fetches the ComponentPreferenceDTO stored against the reference number and the accountId by its version. @param sessionContext @param componentPreferenceDTO @return	This represents the fetchByVersion method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,LoanAccountPaymentInstructionDTO,loanAccountPaymentInstructionDTO „loanAccountPaymentInstructionDTO @return {@link DomainObjectDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule. @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method is responsible for executing Periodic Credit Instructions. @param applicationContext @param accountId @param actualExecutionDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule. @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be responsible for execution of settlement leg of the transaction. This method will invoke {@link SettlementExecutionService#executePreferenceOnline} via an adapter {@link : LoanSettlementApplicationAdapter#executePreferenceOnline} The method works as follows :- <ui><li>Create an instance of LoanSettlementApplicationAdapter.</li> <li>Call the executePreferenceOnline method of the adapter with the required inputs</li> <li>Return {@link SettlementExecutionResponse} which contains information like unsettledAmount , accounting amountTags from settlement module </ui> @param sessionContext @param componentPreferenceDTO @param eventData
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule. @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method is responsible for posting of settlement transaction. For instance, if we try to do backdated installment payment or backdated advance payment then  the amount to be withdrawn from casa account or GL needs to be done via settlement, which is done using this method. @param appContext @param backdatedTransactionDTO @param amountForSettlement @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule. @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	<pre>This method is used for posting the disbursement transaction. Posting the disbursement involves a)Logging Loan Ledger entries for disbursement b)Invoking Settlement Module for credit of the disbursement amount to the beneficiary account (GL , CASA , EXTERNAL) c)Incrementing the Loan balances d)Increasing the disbursement count on the account e)Posting Deductions f)Logging various batch actions (Accelerated Payment , Account Payment Instruction , Advance Alert) e)Invoking interest module for registering the interest plan &#064;param applicationContext &#064;param eventId &#064;param accountId &#064;param disbursementAmount &#064;param feeCollectionDTO &#064;param componentPreferenceDTO @param scheduleIdForPrincipalAmount TODO @param scheduleIdForPeriodicFees TODO @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule. @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	@param applicationContext @param eventId @param accountId @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param sessionContext @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline} This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.   @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This private method will process a single contribution instruction request on theactionDate  for the loan account.  Processing : <ul> <li>First, it will process the componentPreferences(i.e. settlement execution) associated with the  contribution instruction.</li> <li>In case of successful execution of settlement, it will update the contributionAmount present in  LoanAccount via {@link LoanAccount.updateContributionAmount()} and LoanAccountBalances via  {@link LoanBalanceService.updateContributionReceived()}.</li> <li>raise accounting events.</li> <li>raise alerts</li> </ul> @param applicationContext @param loanContributionPayinInstruction @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method will be responsible for execution of settlement leg of the transaction. This method will prepare {@link ComponentPreferenceDTO} and invoke {@link SettlementExecutionService#executePreferenceOnline} via an adapter {@link : LoanSettlementApplicationAdapter#executePreferenceOnline}  @param loanAccountId @param providerAccountId @param transactionAmount @param applicationContext @param loanAccount @param settlementModeType @param eventCode @param string

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	This method is used to perform the drawdown for a loan account. This method would check if drawdown is allowed for loan account, then it calculates the amount to be drawn down, then debit the remitter account and finally credit the loan account. If drawdown is not allowed for any of either loan or remitter account and retry can be put for this drawdown instruction then it puts a drawdown instruction retry. @param applicationContext @param accountDrawdownData
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreference	This method will be used for execution of a existing component preference. This method should be used to execute  preferences as per predefined schedule.  @param accountID @param preferenceID @param productCode @param eventCode @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	This method will be responsible for execution of settlement leg of the transaction. This method will invoke {@link SettlementExecutionService#executePreferenceOnline} via an adapter {@link : LoanSettlementApplicationAdapter#executePreferenceOnline} The method works as follows :- <ui> <li>Create an instance of LoanSettlementApplicationAdapter.</li> <li>Call the executePreferenceOnline method of the adapter with the required inputs</li> <li>Return {@link SettlementExecutionResponse} which contains information like unsettledAmount , accounting amountTags from settlement module </ui> @param sessionContext @param componentPreferenceDTO @param eventData
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param	This method is responsible for posting of settlement transaction. For instance, if we try to do backdated installment payment or backdated advance payment then  the amount to be withdrawn from casa account or GL needs to be done via settlement, which is done using this method. @param appContext @param backdatedTransactionDTO @param amountForSettlement @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			moduleReferenceNo @param messageType @throws FatalException	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	<pre>This method is used for posting the disbursement transaction. Posting the disbursement involves a)Logging Loan Ledger entries for disbursement b)Invoking Settlement Module for credit of the disbursement amount to the beneficiary account (GL , CASA , EXTERNAL) c)Incrementing the Loan balances d)Increasing the disbursement count on the account e)Posting Deductions f)Logging various batch actions (Accelerated Payment , Account Payment Instruction , Advance Alert) e)Invoking interest module for registering the interest plan &#064;param applicationContext &#064;param eventId &#064;param accountId &#064;param disbursementAmount &#064;param feeCollectionDTO &#064;param componentPreferenceDTO @param scheduleIdForPrincipalAmount TODO @param scheduleIdForPeriodicFees TODO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	@param applicationContext @param eventId @param accountId @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO @throws FatalException
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			moduleReferenceNo @param messageType @throws FatalException	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	This method will be responsible for execution of settlement leg of the transaction. This method will prepare {@link ComponentPreferenceDTO} and invoke {@link SettlementExecutionService#executePreferenceOnline} via an adapter {@link : LoanSettlementApplicationAdapter#executePreferenceOnline}  @param loanAccountId @param providerAccountId @param transactionAmount @param applicationContext @param loanAccount @param settlementModeType @param eventCode @param string

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.  @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @throws FatalException	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	This method will be responsible for reversal of settlement leg of the transaction. This method will invoke {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} via an adapter {@link LoanSettlementApplicationAdapter#reverseExecutePreferenceOnline} The method works as follows :- <ul style="list-style-type: none; padding-left: 0;"><li><li>Create an instance of LoanSettlementApplicationAdapter.</li><li>Call the reverseExecutePreferenceOnline method of the adapter with the required inputs</li></li></ul> @param eventCode @param narration

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	<pre>This method is responsible for reversal of disbursement. Reversal of disbursement will involve deleting all data inserted during the disbursement process. &#064;param applicationContext &#064;param accountId &#064;param originalTransactionReferenceNumber @param componentPreferenceId @param isSettlementPayoutReversal @return void
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param sessionContext @param eventCode @param narration
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param sessionContext @param eventCode @param narration @param componentPreferenceId
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseExecutePreferenceOnline	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, {@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting} @param eventCode @param narration	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseUsingComponentPreferenceId	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, <a href="#">{@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting}</a> @param sessionContext @param eventCode @param narration @param componentPreferenceId	<pre>This method is responsible for reversal of disbursement. Reversal of disbursement will involve deleting all data inserted during the disbursement process. &#064;param applicationContext &#064;param accountId &#064;param originalTransactionReferenceNumber @param componentPreferenceId @param isSettlementPayoutReversal @return void
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	reverseUsingComponentPreferenceId	This method will be used for reversal of the settlement instruction. This method will in turn call the settlement service, <a href="#">{@link SettlementExecutionService#reversePreferenceOnlineWithoutAccounting}</a> @param sessionContext @param eventCode @param narration @param componentPreferenceId	This method will be invoked only in case of reversal of first disbursement. @param applicationContext @param loanAccount @param originalTransactionReferenceNumber
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	initializeComponentPreferenceDTO	This method is used to initialize <a href="#">{@link ComponentPreferenceDTO}</a> @param splitType @param moduleType @param directionType @param arrangementId	This method constructs, <a href="#">{@link ComponentPreferenceDTO}</a> required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	initializeComponentPreferenceDTO	This method is used to initialize <a href="#">{@link ComponentPreferenceDTO}</a> @param splitType @param moduleType @param directionType @param arrangementId	This method is used to initialize <a href="#">{@link ComponentPreferenceDTO}</a> @param splitType @param moduleType @param directionType @param commonSettlementDTO @param arrangementId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForSavingsAccount	This method is used to add a CASA(can be any NGP CASA account), mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate	This method constructs, {@link ComponentPreferenceDTO} required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForSavingsAccount	This method is used to add a CASA(can be any NGP CASA account), mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate	This method is used to add a CASA(can be any NGP CASA account), mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param arrangementId @param chequeNo @param chequeDate
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	Narration for event codes: LN_PERIODIC_CREDITThis method is called in backdated reversal of periodic credit. Narration is derived the same way it was logged. @param accountId @param eventId @param transactionReferenceNumber @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method is responsible for executing Periodic Credit Instructions. @param applicationContext @param accountId @param actualExecutionDate
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method constructs, {@link ComponentPreferenceDTO} required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method is used for populating Loan Auto Disbursement Data. @param accountId @param autoDisbursementDate @param bankName @return LoanAutoDisbursementData
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param accountId @param componentPreferenceDTO @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreference	This method return a {@link ComponentPreferenceDTO} created via {@link ComponentPreferenceBuilder} @return ComponentPreferenceDTO	This method is used to perform the drawdown for a loan account. This method would check if drawdown is allowed for loan account, then it calculates the amount to be drawn down, then debit the remitter account and finally credit the loan account. If drawdown is not allowed for any of either loan or remitter account and retry can be put for this drawdown instruction then it puts a drawdown instruction retry. @param applicationContext @param accountDrawdownData

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreferenceDTO	This method fetches the {@link IComponentPreference} via {@link ComponentPreferenceRepository}, converts it to {@link ComponentPreferenceDTO} via assembler and returns {@link ComponentPreferenceDTO} @param accountId @param referenceNumber @return	Narration for event codes: LN_PERIODIC_CREDITThis method is called in backdated reversal of periodic credit. Narration is derived the same way it was logged. @param accountId @param eventId @param transactionReferenceNumber @return
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreferenceDTO	This method fetches the {@link IComponentPreference} via {@link ComponentPreferenceRepository}, converts it to {@link ComponentPreferenceDTO} via assembler and returns {@link ComponentPreferenceDTO} @param accountId @param referenceNumber @return	This method is responsible for executing Periodic Credit Instructions. @param applicationContext @param accountId @param actualExecutionDate
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreferenceDTO	This method fetches the {@link IComponentPreference} via {@link ComponentPreferenceRepository}, converts it to {@link ComponentPreferenceDTO} via assembler and returns {@link ComponentPreferenceDTO} @param accountId @param referenceNumber @return	This method is used for populating Loan Auto Disbursement Data. @param accountId @param autoDisbursementDate @param bankName @return LoanAutoDisbursementData

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchComponentPreferenceDTO	This method fetches the {@link IComponentPreference} via {@link ComponentPreferenceRepository}, converts it to {@link ComponentPreferenceDTO} via assembler and returns {@link ComponentPreferenceDTO} @param accountId @param referenceNumber @return	This method is used to perform the drawdown for a loan account. This method would check if drawdown is allowed for loan account, then it calculates the amount to be drawn down, then debit the remitter account and finally credit the loan account. If drawdown is not allowed for any of either loan or remitter account and retry can be put for this drawdown instruction then it puts a drawdown instruction retry. @param applicationContext @param accountDrawdownData
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForCash	This method is used to add CASH, mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param currency @param cashPositionDTO @param isDenominationTreckingRequired	This method constructs, {@link ComponentPreferenceDTO} required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForCash	This method is used to add CASH, mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param currency @param cashPositionDTO @param isDenominationTreckingRequired	This method is used to add CASH, mode of settlement within {@link ComponentPreferenceDTO} @param priority @param splitValue @param currency @param cashPositionDTO @param isDenominationTreckingRequired
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForExternalAccount	<p>This method is used for Mode of Payment as External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName	This method constructs, {@link ComponentPreferenceDTO} required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForExternalAccount	<p>This method is used for Mode of Payment as External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName	<p>This method is used for Mode of Payment as External Account.  It is being preferred when Payment is expected to be paid by External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForPaymentToExternalAccount	<p>This method is used for Mode of Payment as External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName	This method constructs, {@link ComponentPreferenceDTO} required for interaction with the settlement module. @param fromAccountNo @param toAccountNo @param chequeNo @param chequeDate @param currencyDenominationDetails @param amount @param directionType @param settlementModeType @return componentPreferenceDTO
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	addModeForPaymentToExternalAccount	<p>This method is used for Mode of Payment as External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName	<p>This method is used for Mode of Payment as External Account.   It is being preferred when Payment is to be paid to External Account. </p> @param priority @param splitValue @param transactionCurrency @param externalAccountNumber @param creditorName
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchInternalComponentPreference	@param componentPreferenceDTO @return	
ILoanSettlementApplicationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanSettlementApplicationAdapter	fetchInternalComponentPreference	@param componentPreferenceDTO @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will be used to forcefully reject on the successfully processed incoming transaction due to various business transaction. This will pass accounting entry and reverse the IMC calls. It will mark the transaction as rejected/authorised. Remaining Accounting entries will be passed as part of processReturn method. It will mark the initial state of the transaction as REJECTED as well as the current state. @param txnContext @param type @param collectionType @param rejectCode @param rejectCode @param primaryReasonCode @param secondaryReasonCode @param originalTransactionLog @param messageLogItem TODO
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will undo the credit of uncleared funds to the account. It will be required if the outbound cheque against which the funds was earlier credited, has now been dishonored @param transactionLog TODO @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will notify the account owning module about the succes or failure of the transaction that was initiated by that account. @param sessionContext @param transactionLog @param transactionEventType @return @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will allow the system to reverse the transaction that has already been submitted and dispatched. Reversal is usually done by bank staff, when the mistake is done by staff. No penalty interest is charged here. @param transactionContext @param fundExchangeType @param collectionType @param transactionReferenceNo @param subSequenceNumber @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyRejectedItem	@param accountTransactionDTO @param direction @return @throws FatalException	@param transactionLog @param messageLogItem @param accountTransactionDTO @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method wil withdraw funds from an account of any type. It will internally first determine the type of account before delegating the call to the module adapter @param accountTransactionDTO @param messageIdentification @param accountID Indicates account identifier @param log Indicates the entire transaction log @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	@param accountTransactionDTO @param messageIdentification @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will transfer funds to the given account. The given account can be of any module. @param accountID Indicates the account identifier @param amount Indicates the amount of funds to be transferred @param log Indicates the transaction log @return
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyProcessedItem	@param accountTransactionDTO @param direction @return @throws FatalException	@param accountTransactionDTO @param messageIdentification @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	This service validate the policy of immediate credit as well as perform the Immediate credit on provided data. It will raise an accounting event for the credit and also audit the transaction @throws FatalException @see com.ofss.fc.app.pc.core.IOutgoingCollectionService#doImmediateCredit(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.pc.dto.log.InstrumentTransferTransactionLogDTO, com.ofss.fc.app.pc.dto.log.InstrumentLogItemDTO, com.ofss.fc.datatype.Money)
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	This method will do the funds realization associated with the instrument based on the event, and nature of transaction. @fcb.param transactionLog @fcb.param transactionEventType @fcb.param messageLogItem

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	notifyFundsReleasedForItem	@param accountTransactionDTO @param direction @return @throws FatalException	It it undo the transaction and raise dishonor accounting. @param transactionContext @param transactionLog @param messageLogItem @return @throws FatalException
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	@param logItem @param amount @param txnlog
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	This method will be called before a debit or a credit transaction is processed for a CASA account, to validate the account status for credit/debit. @param amount @param transactionReferenceNo @param messageIdentification @param serviceId @param eventId @param sessionContext @param accountId @param accountingEntryType
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	This method validates the loan account.
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	This method is used for determining the fee and levy fee for returns processing. @param inputTransactionLog @param instrumentType @param copiedTxnLog @param eventId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	<p>&lt;p&gt;This method will validate the account before trying to debit the account. It will delegate the call to the appropriate module to which the account belongs &lt;/p&gt; &lt;p&gt; First , the system will check if the account type is available or not in the transaction log. &lt;br&gt; If not then, the system will determine which module does the given account belongs to by &lt;/p&gt; &lt;ul&gt; &lt;li&gt;making a call to fetchAccountByAccountType(SessionContext, String, AccountType)&lt;/li&gt; &lt;li&gt;From the fetched dto get the account type.&lt;/li&gt; &lt;li&gt;Converting the Account type to TaskModuleType&lt;/li&gt; &lt;li&gt;Update the log with the TaskModuleType&lt;/li&gt; &lt;/ul&gt; &lt;p&gt; and then call the appropriate module adapter based on the TaskModuleType &lt;/p&gt; @param transactionContext @param bankDraft @param settlementModeType</p>
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountID @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	@param transactionContext @param eventId @param iBankDraft @param mode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountId @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	<p>This method will validate the account before trying to debit the account. It will delegate the call to the appropriate module to which the account belongs</p> <p>First , the system will check if the account type is available or not in the transaction log. If not then, the system will determine which module does the given account belongs to by</p> <ul style="list-style-type: none"> <li>making a call to <a href="#">com.ofss.fc.app.account.service.core.AccountInquiryApplicationService.fetchAccountByAccountType(SessionContext, String, AccountType)</a></li> <li>From the fetched dto get the account type.</li> <li>Converting the Account type to TaskModuleType</li> <li>Update the log with the TaskModuleType</li> </ul> <p>and then call the appropriate module adapter based on the TaskModuleType</p> <p>@param transactionContext @param accountId @param transactionLog @param messageLogItem @return @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	verifyAccount	@param accountId @param amount @param transactionReferenceNo @param messageIdentification @param direction @param serviceId @param eventId @return	<p>This method will validate the account before trying to credit the account. It will delegate the call to the appropriate module to which the account belongs</p> <p>First, the system will check if the account type is available or not in the transaction log. If not then, the system will determine which module does the given account belongs to by</p> <ul style="list-style-type: none"> <li>making a call to <code>com.ofss.fc.app.account.service.core.AccountInquiryApplicationService.fetchAccountByAccountType(SessionContext, String, AccountType)</code></li> <li>From the fetched dto get the account type.</li> <li>Converting the Account type to <code>TaskModuleType</code></li> <li>Update the log with the <code>TaskModuleType</code></li> </ul> <p>and then call the appropriate module adapter based on the <code>TaskModuleType</code></p> <p>@param transactionContext @param accountId @param transactionLog @param messageLogItem @return ValidationResultDTO @throws FatalException</p>
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	This service allows the value date to be incremented by specific days. @fcb.param context @fcb.param type @fcb.param collectionType @fcb.param transactionReferenceNo @fcb.param messageId @throws Exception
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	This service allows calculate the difference between value dates and call the extend float By Days service @fcb.param sessionContext @fcb.param transactionLog @fcb.param messageLogItem @fcb.param valueDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	
IPaymentsNotificationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPaymentsNotificationAdapter	updateDateOnFloatExtension	@param accountId @param newValueDate @param referenceNumber @param subSequenceNumber @throws FatalException	
IPProductGroupTemplateProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPProductGroupTemplateProcessAdapter	startProcess	This method initiates the ProductGroupTemplateProcess @param productGroup String	
IPProductGroupTemplateProcessAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPProductGroupTemplateProcessAdapter	startProcess	This method initiates the ProductGroupTemplateProcess @param productGroup String	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILoanSecurityAdapter	fetchAccountByAccountType		Method to get the Loan Outstanding Balance. @throws FatalException
ILoanSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\ILoanSecurityAdapter	fetchAccountByAccountType		
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the cheque deposited by him has been presented in outward clearing and the funds have been credited to his account @param activityId @param accountNumber @param eventId @param chequeNumber @param valueDate @param chequeAmount @param dateOfTransaction @param transactionReferenceNum
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the cheque issued by him has been presented in Inward clearing and his account has been debited @param activityId @param accountNumber @param eventId @param chequeNumber @param valueDate @param chequeAmount @param dateOfTransaction @param transactionReferenceNumber
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the cheque deposited by him has been returned by the paying bank @param activityId @param accountNumber @param eventId @param chequeNumber @param valueDate @param chequeAmount @param dateOfTransaction @param currencyCode @param rejectReason @param partyId @param netDebitAmount @param feeCurrencyCode @param feeAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the cheque issued by him has been returned to the presenting bank. @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction @param currencyCode @param rejectReason @param partyId
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the Incoming Collection transaction initiated by him has been rejected @param activityId @param eventId @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode @param instructionNumbar @param counterpartyName @param counterpartyBank @param txnAmount @param @param
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that the Incoming Payment transaction initiated by him has been rejected
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify the debit transaction in the account @param activityId @param eventId @param debitAccountNo @param customerName @param transactionAmount @param transactionDate @param conterpartyName @param conterpartyBank @param conterpartyAccountNo @param currencyCode @param rejectReason @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify the credit transaction in the account @param activityId @param eventId @param debitAccountNo @param customerName @param transactionAmount @param transactionDate @param conterpartyName @param conterpartyBank @param conterpartyAccountNo @param currencyCode @param rejectReason @param partyId
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	Payment rejection alert is sent to the customer when the payment is rejected by the beneficiary bank due to reasons such as incorrect data, account blocked etc. The rejection alert is sent with the specific reject reasons for which codes are maintained in the system @param activityId @param eventId @param debitAccountNo @param customerName @param transactionAmount @param transactionDate @param conterpartyName @param conterpartyBank @param conterpartyAccountNo @param currencyCode @param rejectReason @param partyId
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method persists activity and event log for given activity and event id @param activityId @param eventId @param activityLog @param eventDate @throws FatalException
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert for Linked account setup,linked account verification success, Linked account verification failed, start direct debit and stop direct debit. Alerts will get generate as per the activity and event raised. @param activityId @param eventId @param debtorBank @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert for Outgoing collection success,Outgoing collection failed, Incoming collection success , Incoming Collection return, Dishonour of outgoing collection and Outward collection honoured.Alets will get generate as per the activity and event raised. @param activityId @param eventId @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode @param instructionNumbar @param counterpartyName @param counterpartyBank @param txnAmount @param transactionDate
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	@param activityId @param eventId @param customerId @param customerName @param debtorAccountNo @param creditorAccountNo @param currencyCode @param instructionNumbar @param rejectReason @param counterpartyName @param txnAmount @param transactionDate @param counterpartyBank @param productId
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This alert is sent to the customer to notify that periodic bill payment has been done and the funds have been debited from his account @param activityId @param eventId @param custAccountId @param customerId @param transactionAmount @param currencyCode @param billerAccountId @param transactionDate @param bankName
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method is used to send an alert to customer to notify him that bank draft print against the information provided. @param activityId @param customerId @param eventId @param accountId @param instrumentAmount @param beneficiaryName @param instrumentStatus @param processingDate @param instrNumber @param serialNumber
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method triggers notification event. @param partyId @throws FatalException
IPCEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCEventProcessingAdapter	processAdhocAlert	send adhoc alerts to payee @param adhocAlertDTO @throws FatalException	
IPartyInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyInterestAdapter	partyIdMaintenance		If there is a change in primary ownership of account then it will update the Interest module for the change in ownership.  @param accountId @param newPrimaryPartyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	linkModuleBranchTransfer		This method will be used to transfer account from one branch to other. Steps need to be followed: Null check for all mandatory fields. common business policy will be validated. Based on the module type whether CASA, TD or Loan, respective API will be called through adapter. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, ,transactionStatus This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,AccountBranchTransferBusinessPolicyDTO,accountBranchTransferDetailsDTO, TODO: Explain the @return Transaction status @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	linkModuleBranchTransfer		Method linkModuleBranchTransfer is being used to decide the service call based on moduleType. ie if account type is CASA call method linkDDAccountToOtherBranch(sessionContext, ddAccountBranchTransferDetailsDTO) Method to link Module wise Account Branch Transfer.
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	Method will return account branch code. @param accountBranchTransferDetailsDTO @return
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method will return the facility linked to the CASA or OD account. @param sessionContext @param accountId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method will return the facility linked to the LOAN account. @param sessionContext @param accountId @return @throws FatalException
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method is used for pagination purpose. It fetches the fixed number of arrangement records from DB based on the callcount. so if fixed page size is say 50, and call count is 1 , then it fetches records from 0-50. if call count is 2 then it fetches records from 51-100 and so on. It then calls respective modules to get other related data from them. in case of singleCustomerPortfolioTransfer when partyId is given, the respective modules is called directly to fetch the accountIds based on partyID and then from agentArrangementLinkage DB the corresponding commission Receipt Details are fetched.
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method is used to fetch the Details which the user has queried for a particular module. First the queries submissinIds and details are fetched from CASA module and then agentArrangementLinkage DB is queried to get commission Receipt Id and Name @param sessionContext @return @throws FatalException
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method will consume OFSAA request.It will simply create a response object and return. @param dto @return pricingResponse
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	This method will be used for inquiring as well as maintaining economic cost details. If economic cost is already maintained and valid then it will return the details in response and if same is not maintained or not valid then it will be created first and then will be returned.  Steps are as follows -   <ol><li>load demanddepositAccount and validate status, fixed period and economic levy flag. <li>load valid economic cost through {@link EconomicCostQuoteRepository}. <li>If quote is present then check - if quote is expired then regenerate. if quote outstanding amount and new utilized OD not same then regenerate quote. <li>if quote is not present then regenerate it. <li>while generating Call OFSAAFTP system to inquire Economic cost through {@link IDDAFTPAdapter#fetchEconomicCost(String, Money)}. <li>persist generated quote and return the same. </ol> @param accountId @param utilizedAmount This field denotes amount for which quote will be generated. In case of casa fixed period OD , it will be utilized OD before transaction @throws FatalException
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountDetails	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse	Method will be used to return account details based on Account type. @param accountId @param accountType @return AccountInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountByAccount	This method fetched account details only on basis of account Id @param sessionContext @param accountId @return	<p>&lt;p&gt;This service searches for a party given a set of search criteria.&lt;br/&gt;This service will search for parties from various sources depending on the adapter that is implemented for &lt;b&gt; {@link com.ofss.fc.common.integration.config.IntegrationAdapters#party_service_core_PartySearchApplicationService_searchParty} &lt;/b&gt;. &lt;br/&gt; This adapter dictates whether internal or external source is responsible for getting details for the search results. NullAdapter is implementation of this interface where integration end points are absent. The input search criteria {@link com.ofss.fc.app.search.dto.SearchCondition#searchType} defines the type of party search intended. Any one of the following values can be supplied in this variable: &lt;dl&gt; &lt;dt&gt;PrimarySearchDetails&lt;/dt&gt; &lt;dd&gt;For basic search conditions like partyId, partyType, partyClass (Refer {@link com.ofss.fc.domain.party.service.dto.search.PrimarySearchDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;IdentificationDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of identification details like identificationType, identificationValue, issueDate, expiryDate (Refer {@link com.ofss.fc.domain.party.service.dto.search.IdentificationDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;NameDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of secondary name details like firstName, middleName, shortName, lastName (Refer {@link com.ofss.fc.domain.party.service.dto.search.NameDetails} for complete list of input criteria)&lt;/dd&gt; &lt;dt&gt;AddressDetails&lt;/dt&gt; &lt;dd&gt;For search on the basis of address details like city, state, country (Refer {@link com.ofss.fc.domain.party.service.dto.search.AddressDetails} for complete list of input criteria)&lt;/dd&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p><b>BrokerDetails</b> For search on the basis of broker details like spid, serviceProviderStatusType, commissionRecipientId, commissionRecipientName (Refer <a href="#">{@link com.ofss.fc.domain.party.service.dto.search.BrokerDetails}</a> for complete list of input criteria)</p> <p><b>RoleDetails</b> For search on the basis of role details like roleId (Refer <a href="#">{@link com.ofss.fc.domain.party.service.dto.search.RoleDetails}</a> for complete list of input criteria)</p> <p>For each of the above search conditions, the user can choose whether the search is to use exact match or not by specifying <a href="#">{@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.isExactSearch}</a>. <i>Default value is true</i></p> <p>For example if the user intends to search on first name, and specifies input as <code>isExactSearch=true</code> and <code>firstName='A'</code>, the service will fetch records where the <code>firstName='A'</code> but if <code>isExactSearch=false</code>, the service will fetch records where the <code>firstName</code> like <code>'%a%'</code></p> <p>This transaction supports the Pagination feature. The user can control the number of records to be fetched by specifying <a href="#">{@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.totalRecordsFetched}</a> <i>Default value is 10</i></p> <p>The user can also specify the first row from which data is sought by setting <a href="#">{@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.firstResult}</a>. <i>Default value is 0</i></p> <p>The party search results returned in a search contain the basic details of each party found <a href="#">{@link com.ofss.fc.app.party.dto.search.output.PartySearchResponseDTO}</a>. There may be times when additional details are required. To facilitate this, the search</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>transaction supports an inquiry level which the user can specify through <code>{@link com.ofss.fc.app.party.dto.search.input.PartySearchRequest.isBasicSearch}</code> <i>&lt;i&gt;Default value is true&lt;/i&gt;</i></p> <p>The response fetches the following details for parties which match the search criteria:</p> <ul style="list-style-type: none"> <li>Party id - unique identifier for the party</li> <li>Party name - full name of the party</li> <li>Party class</li> <li>Party type - indicator of whether the party is Individual, Bank, Trust etc.</li> <li>Date of birth - birth date of the party</li> <li>Email id - email of the party</li> <li>Postal code - address postal code of the party</li> <li>Role id - customer or banker role</li> </ul> <p>@param sessionContext @param partySearchrequest @return @throws FatalException</p>
IAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAdapter	fetchAccountByAccount	This method fetched account details only on basis of account Id @param sessionContext @param accountId @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	This method is designed to raise a request for Account Closure. For further details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosureApplicationService#createClosureRequest(SessionContext, DDAClosureRequestDTO)}
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	addSettlementDetails	This method add ComponentPreferenceDTO. This method pass ComponentPreferenceDTO to settlement module, which have the information about the beneficiary accounts. In return settlement module returns a reference number which needs to be stored in DDA module. @param componentPreferenceDto @param eventId @param allowedSplitTypes @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	updateSettlementDetails	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	updateSettlementDetails	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	updateSettlementDetails	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	updateSettlementDetails	This method updates the ComponentPreferenceDTO stored against the reference number. @param componentPreferenceDto @param eventId @param allowedSplitTypes	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method is to prepare instance log for failure executions. @param account @param instructionRequest @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method prepares the ComponentPreferenceKeyDTO with account Id and component Preference Id and inquire from settlement for component preference then it returns that to caller method. @param accountId @param componentPreferenceId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method gives settlement payout GL for given debit instruction. While executing it has given credit on that settlement payout GL. @param debitInstruction @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method is designed to fetch the details about the Account Closure Request, which is already being raised. For further details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosureApplicationService#fetchClosureRequest(SessionContext, DDAClosureRequestDTO)}
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method is designed to fetch the details about Account Closure. For further Details: {@link com.ofss.fc.app.dda.service.account.core.IDDAClosureApplicationService#fetchClosureDetails(String, int)}
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method provides the settlement details of all Accounts who are using the provided account as Beneficiary for Account Closure. It would return the total number of Settlements for Account Closure, in which the provided Account has been used as Beneficiary by other Accounts. If the number of Settlements for Account Closure are found, it would raise an error, and will not permit to close the account. @param accountId @param eventCode
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method validates that contribution should be allowed only for settlement mode types DDA/GI. If its any other settlement mode other than the one that are specified above , then error will be thrown.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	Execution of Periodic Funds Transfer instructions (Standing Instruction) is handled at batch process, Beginning of Day operations (BOD). The execution of Periodic Funds Transfer instructions happens on the previous working day, or the next working day in case the execution day is a holiday, This is the concrete implementation of the execute method. This method executes the Standing Instruction as follows: <ol><li>The Periodic Transfer Instruction will always be initiated during BOD.</li> <li>The system checks the accounts which are due for Periodic Transfer based on the Periodic Transfer Start Date and the Recurrence pattern maintained.</li> <li>The System Validates end date for the instruction. The Conditions when the Periodic Transfer Instruction is to be terminated is maintained in the recurrence pattern.</li> <li>System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account</li> <li>The system validates the holiday processing rules (Refer Holiday Processing)</li> <li>System identifies the accounts eligible to execute Periodic Transfer Instruction on this processing date</li> <li>The System Checks If the Funds are available in the Provider accounts</li> <li>The Execution of the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>instruction is based on the priority defined for the accounts.</p> <ul style="list-style-type: none"> <li>The System Executes the transfers</li> <li>The system updates the book balance and passes the accounting entries (posting dated will be the current system date)</li> <li>The System maintains a Transaction Reference number for Audit purpose</li> <li>The System updates the Next Execution date. (For Accounts, which the Periodic Transfer Instruction ends the next execution date is not updated)</li> </ul> <p><b>Holiday Processing for Standing Instructions</b></p> <ul style="list-style-type: none"> <li>The Holiday Processing can configured for the <math>n^{\text{th}}</math> day of execution <math>n^{\text{th}}</math> as defined by the business rules of the bank</li> <li>If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday</li> <li>If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day.</li> <li>Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days.</li> </ul> <ul style="list-style-type: none"> <li><b>Scenario 1: <math>n^{\text{th}}</math> Day Of Execution <math>n^{\text{th}}</math> is <math>n^{\text{th}}</math> Previous Day <math>n^{\text{th}}</math></b></li> <li>The System identifies that the next day is a holiday in the Bank Calendar</li> <li>The System validates all the account due for sweep out till the next working day.</li> <li><b>Scenario 2: <math>n^{\text{th}}</math> Day Of Execution <math>n^{\text{th}}</math> is the <math>n^{\text{th}}</math> Next Working Date <math>n^{\text{th}}</math></b></li> <li>The System checks if the previous day was a holiday in the Bank Calendar</li> <li>The System validates all the account due for sweep out from the last working date.</li> </ul> <p>This method works as follows :</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>generate transactionReferenceNumber. {@link UUIDGenerator#generate()} This method works as follows : &lt;br&gt; &lt;pre&gt; Event Id = STANDING_EVENT_CODE boolean isAllowed = true; List&lt;InstructionInstance&gt; instructionInstanceList; &amp;lt;i&gt;First, it will check whether this instruction execution is allowed or not. If it is not allowed, then the SI is treated to be a failed instruction.&amp;lt;/i&gt;; try { Call {@link IDemandDepositAccount#validateStatus()}. } catch (Exception e) { &amp;lt;i&gt;Log the message, and execute next SweepOut Instruction&amp;lt;/i&gt;; isAllowed = false; } &amp;lt;i&gt;If the validate status allows the instruction to be processed, then the actual instruction execution is processed.&amp;lt;/i&gt;; if(isAllowed) { try { instructionInstanceList = call {@link IDDASettlementAdapter#processSettlement()} pass requestedSweepOutAmount. } catch(Exception e) { &amp;lt;i&gt;This may occur if the beneficiary account don't allow credit to occur.&amp;lt;/i&gt;; &amp;lt;i&gt;In this case also, the instruction is treated as a failed instruction&amp;lt;/i&gt;; isAllowed = false; } } if instruction execution failed if(!isAllowed) { check if(retriesCount &lt; maxRetries) { if(retriesCount == 0) actualExecutionDate = {@link RecurrencePatter#calcNexExecutionDate()} nextExecutionDate = next working day of the bank. retriesCount++; &amp;lt;i&gt;Since a failed instruction is to be processed again, if maxRetries count permits.&amp;lt;/i&gt;; } else (retriesCount == maxRetries) { ignore failed instruction. reset retriesCount to 0. nextExecutionDate = actualExecutionDate. &amp;lt;i&gt;If maxRetries count don't allow an instruction to be processed again, the instruction will be ignored.&amp;lt;/i&gt;; } } if(actualExecutionDate ==</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>nextExecutionDate) { &amp;lt;i&amp;gt;If an instruction execution due to its failure, and the actual next instruction execution falls on the same day, then the failed instruction will be ignored. Only one SI will be processed on a given day.&amp;lt;/i&amp;gt; reset retriesCount to 0. ignore failed instruction. }</p> <p>subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate  executionDate = pickupDate;  subInstructionNumber++; Log Instruction instance for the execution of the instruction. } copy nextExecutionDate to lastExecutionDate. copy pickupDate to lastPickupDate calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update StandingInstruction by calling {@link StandingInstructionRepository#updateStandingInstruction()} Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link StandingInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link StandingInstructionRequest#eventCode} Prepare {@link TransactionAmount} object from the standing instruction amount. {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> StandingInstructionRequest#transactionBranch} {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; @param instructionRequest </pre>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method will derive the funding strategy applicable for the operation. It will load the settlement for the modes if the mode is DDA then it is DemandDeposit type of funding strategy else it is Clearing type of strategy. @return FundingStrategy
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	This method is to prepare instance log for failure executions. @param account @param instructionRequest @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	Method will get the comma separated string of all the account id in the component preferences.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	Method will get the comma separated string of all the account id in the component preferences.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	inquireSettlementDetails	This method inquires the ComponentPreferenceDTO stored against the reference number and the account id. @param componentPreferenceDto @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is exposed to perform delete operation for SweepOutInstruction. This method works as follows :   <pre> sweepOutInstruction = Call {@link SweepOutInstructionAssembler#toDomainObject()} call {@link ISweepOutInstructionDomainService#deleteSweepOutInstruction()} call {@link ISweepOutInstructionDomainService#fetchSweepOutInstruction()} &lt;li&gt; to get settlement reference number which need to be passed to settlement adapter.&lt;/li&gt; Get the instance of the settlement adapter {@link IAdapter#getAdapter()} &lt;i&gt;This method returns the instance of {@link IDDASettlementAdapter} Call settlement adapter to delete beneficiary account details. &lt;li&gt;{@link IDDASettlementAdapter#deleteSettlementDetails()} Pass {@link ComponentPreferenceDTO} object, along with reference number&lt;/li&gt; </pre> @param sessionContext MI,SessionContext,sessionContext, ,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param sweepOutInstructionDTO MI, {@link SweepOutInstructionDTO}, sweepOutInstructionDTO, &lt;/br&gt; It contains the following attributes: &lt;ol&gt; &lt;li&gt;key: {@link InstructionKeyDTO} MI. It should contain the accountId and the InstructionNumber&lt;/li&gt; &lt;/ol&gt; @return {@link TransactionStatus}. The response contains the Transaction Status. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;/ul&gt; &lt;ol&gt; &lt;li&gt;{@link CommonDDAAdapterErrorConstants#ERROR_IN_DDA_SETTLEMENT_INTERACTION}&lt;/li&gt; &lt;li&gt;{@link InstructionDomainConstant#ERROR_IN_INSTRUCTION_DELETION}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ol&gt;</p>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This override method will do the action. Action here is to delink the SweepIn Facility.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deleteSettlementDetails	This method removes the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	The method will be used for updating the closure request details. It will primarily be used to update the retention strategy by the relationship manager. The other details cannot be updated once the request is submitted. <pre>The method works as follows : <ul style="list-style-type: none"> <li>Call the repository method {@link DDAClosureRepository#getInstance} to create an instance of the repository.</li> <li>Call the method {@link DDAClosureRepository#update} to update the closure request details.</li> <li>If the retention strategy is proceed with closure request, then the request needs to be sent to the BEPL worklist.</li> <li>ElseIf the retention strategy is reject closure request, then</li> <li>Restore the original status by calling the method {@link AccountStatusManagementService#undoMarkedClosureStatus}.</li> <li>Call the repository method {@link DDAClosureRepository#close} to close the closure request.</li> <li>Check the overridable conditions by calling the method {@link OverridableFinancialBusinessPolicy#validate}</li> </ul> @param closureRequest

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	closeSettlementDetails	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is exposed to perform close operation for StandingInstruction @param standingInstruction
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	closeSettlementDetails	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	<p>This is the concrete implementation of the execute method. This method executes the Sweepout Instruction as follows:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;The Sweep out is initiated during End Of Day Batch processing if the EpochType is set as EOD, if it is set up as BOD, it initiates in Beginning of the Operation Day.&lt;/li&gt;</li> <li>&lt;li&gt;The system checks the accounts which are due for sweep out based on the Sweep out Start Date and the Recurrence pattern maintained.&lt;/li&gt;</li> <li>&lt;li&gt;The System Validates end date for the instruction. The Conditions when the Sweep out is to be terminated is maintained in the recurrence pattern.&lt;/li&gt;</li> <li>&lt;li&gt;System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account&lt;/li&gt;</li> <li>&lt;li&gt;The system checks if the previous calendar date was a holiday, refer holiday processing explained in the documentation.&lt;/li&gt;</li> <li>&lt;li&gt;System identifies the accounts eligible to execute Sweep outs on this processing date&lt;/li&gt;</li> <li>&lt;li&gt;The System Checks the following balances defined for the sweep out arrangement.&lt;/li&gt;</li> </ul> <ul style="list-style-type: none"> <li>&lt;li&gt;The Cash Balance ( Minus the Hold)&lt;/li&gt;</li> <li>&lt;li&gt;Minimum amount To be retained&lt;/li&gt;</li> <li>&lt;li&gt;Threshold Balance&lt;/li&gt;</li> <li>&lt;li&gt;Maximum Sweep Out amount&lt;/li&gt;</li> <li>&lt;li&gt;The Amount Basis to arrive at the sweep out amount.&lt;/li&gt;</li> </ul> <ul style="list-style-type: none"> <li>&lt;li&gt;The Execution of the instruction is based on the level and the priority defined for the accounts.&lt;/li&gt;</li> <li>&lt;li&gt;The System Executes the transfers- Amounts which are greater than the maximum limit</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>specified in the sweep out instruction are swept out to a pre-specified Savings account or a Time Deposit account.</p> <ul style="list-style-type: none"> <li>The System Sweeps out funds based on the Sweep Out Amount Basis Defined during Set up</li> <li>The system updates the book balance and passes the accounting entries (posting dated will be the current system date)</li> <li>The System maintains a Transaction Reference number for Audit purpose</li> <li>The System updates the Next Execution date . (For Accounts which the Sweep out Instructions ends the next execution date is not updated.</li> <li>The system executes the next instruction in the same way as the first instruction execution.</li> </ul> <ul style="list-style-type: none"> <li><b>Holiday Processing for Sweep Out Instructions</b></li> <li>The Operations Executive can configure the day of execution as defined by the business rules of the bank</li> <li>If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday</li> <li>If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day.</li> <li>Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days.</li> </ul> <ul style="list-style-type: none"> <li><b>Scenario 1: Day Of Execution is Previous Day</b></li> <li>The System identifies that the next day is a holiday in the Bank Calendar</li> <li>The System validates all the account due for sweep out till the next working day.</li> </ul> <ul style="list-style-type: none"> <li><b>Scenario 2: Day Of Execution is the Next Working Date</b></li> <li>The System checks if the previous day was a holiday in the Bank Calendar</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;li&gt;The System validates all the account due for sweep out from the last working date.&lt;/li&gt;&lt;/li&gt; &lt;/ol&gt;</p> <p>&lt;/ul&gt; For Example: &lt;ol&gt; &lt;li&gt;&lt;b&gt;Single provider Single beneficiary&lt;/b&gt; &lt;ul&gt; &lt;li&gt;The Cash Balance ( Minus the Hold) = 25000&lt;/li&gt; &lt;li&gt;Overdraft Balance (If considered) = NA&lt;/li&gt; &lt;li&gt;Minimum amount To be retained = 5000&lt;/li&gt; &lt;li&gt;Threshold Balance= NA&lt;/li&gt; &lt;li&gt;Maximum Sweep Out amount = 15000&lt;/li&gt; &lt;li&gt;Amount basis: Percentage (25%)&lt;/li&gt; &lt;li&gt;The sweep out amount = (Cash Balance <math>\cap</math> <math>\Downarrow</math> Minimum amount to be retained)Amount basis&lt;/li&gt; &lt;li&gt;=(25000-5000)25% =5000&lt;/li&gt; &lt;li&gt;Since 5000 is less than 15000 (The Max Sweep out Amount ) the amount swept out is 5000.00&lt;/li&gt;&lt;/li&gt; &lt;/ul&gt;</p> <p>&lt;li&gt;&lt;b&gt;Single Provider Multiple Beneficiary&lt;/b&gt; &lt;ul&gt; &lt;li&gt;The Cash Balance ( Minus the Hold) = 25000&lt;/li&gt; &lt;li&gt;Overdraft Balance (If considered) = NA&lt;/li&gt; &lt;li&gt;Minimum amount To be retained = 5000&lt;/li&gt; &lt;li&gt;Threshold Balance= NA&lt;/li&gt; &lt;li&gt;Maximum Sweep Out amount = 15000&lt;/li&gt; &lt;li&gt;Amount basis: Multiples of 400&lt;/li&gt; &lt;ul&gt; &lt;li&gt;Beneficiary A <math>\cap</math> <math>\Downarrow</math> 20%&lt;/li&gt; &lt;li&gt;Beneficiary B <math>\cap</math> <math>\Downarrow</math> 50%&lt;/li&gt; &lt;li&gt;Beneficiary C <math>\cap</math> <math>\Downarrow</math> 30%&lt;/li&gt; &lt;/ul&gt; &lt;li&gt;The Sweep Out Amount = (25000-5000) = 15000&lt;/li&gt; &lt;li&gt;Since the amount basis is multiples of 400 only 14800 (40037 times) is eligible to be swept out&lt;/li&gt; &lt;ul&gt; &lt;li&gt;Beneficiary A = 1480020% = 2960&lt;/li&gt; &lt;li&gt;Beneficiary B = 14800 50%= 7400&lt;/li&gt; &lt;li&gt;Beneficiary C = 14800 30% = 4440&lt;/li&gt; &lt;/ul&gt; &lt;li&gt;If any sweep out instruction fails the amount will remain with the provider account. For instance the sweep out to <math>\cap</math> <math>\Downarrow</math> Beneficiary B<math>\cap</math> <math>\Downarrow</math> Fails, then only 7440 ( to A- 2960 + to C 4440) is debited from the provider account.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ol&gt; &lt;br&gt;This method works as follows : &lt;br&gt; &lt;pre&gt; &lt;pre&gt; Event</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> Id = SWEEP_OUT_EVENT_CODE boolean isAllowed = true; try { Call {@link IDemandDepositAccount#validateStatus()}. }catch (Exception e) { &amp;lt;i&amp;gt;Log the message, and execute next SweepOut Instruction&amp;lt;/i&amp;gt; isAllowed = false; } eligibleSweepOutAmount = {@link SweepOutInstruction#deriveEligibleSweepOutAmount() generate transactionReferenceNumber {@link UUIDGenerator#generate() List&amp;lt;InstructionInstance&amp;gt; instructionInstanceList; for( {@link BeneficiaryAccount} beneficiaryAccount : {@link SweepOutInstruction#getBeneficiaryAccountList()}) { requestedSweepOutAmount = {@link SweepOutInstruction#deriveRequestedSweepOutAmount( eligibleSweepOutAmount)} if(requestedSweepOutAmount &amp;lt;= eligibleSweepOutAmount) { if(isAllowed) { instructionInstanceList = call {@link IDDASettlementAdapter#processSettlement()} pass requestedSweepOutAmount. } } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate executionDate = pickupDate; subInstructionNumber++; Log Instruction instance for the execution of the instruction. } calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update SweepOutInstruction by calling {@link SweepOutInstructionRepository#updateSweepOutInst ruction()} &amp;lt;i&amp;gt;Fee will be charged on every sweep out instruction execution, ie. irrespective of the sub instructions, fee management service will be called </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> only once.&lt;i&gt; Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link SweepOutInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link SweepOutInstructionRequest#eventCode} {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link SweepOutInstructionRequest#transactionBranch} TODO {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; @param instructionRequest </pre>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	closeSettlementDetails	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	Execution of Periodic Funds Transfer instructions (Standing Instruction) is handled at batch process, Beginning of Day operations (BOD). The execution of Periodic Funds Transfer instructions happens on the previous working day, or the next working day in case the execution day is a holiday, This is the concrete



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>implementation of the execute method. This method executes the Standing Instruction as follows: &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;The Periodic Transfer Instruction will always be initiated during BOD.&lt;/li&gt;</li> <li>&lt;li&gt;The system checks the accounts which are due for Periodic Transfer based on the Periodic Transfer Start Date and the Recurrence pattern maintained.&lt;/li&gt;</li> <li>&lt;li&gt;The System Validates end date for the instruction. The Conditions when the Periodic Transfer Instruction is to be terminated is maintained in the recurrence pattern.&lt;/li&gt;</li> <li>&lt;li&gt;System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account&lt;/li&gt;</li> <li>&lt;li&gt;The system validates the holiday processing rules (Refer Holiday Processing)&lt;/li&gt;</li> <li>&lt;li&gt;System identifies the accounts eligible to execute Periodic Transfer Instruction on this processing date&lt;/li&gt;</li> <li>&lt;li&gt;The System Checks If the Funds are available in the Provider accounts&lt;/li&gt;</li> <li>&lt;li&gt;The Execution of the instruction is based on the priority defined for the accounts.&lt;/li&gt;</li> <li>&lt;li&gt;The System Executes the transfers&lt;/li&gt;</li> <li>&lt;li&gt;The system updates the book balance and passes the accounting entries (posting dated will be the current system date)&lt;/li&gt;</li> <li>&lt;li&gt;The System maintains a Transaction Reference number for Audit purpose&lt;/li&gt;</li> <li>&lt;li&gt;The System updates the Next Execution date. (For Accounts, which the Periodic Transfer Instruction ends the next execution date is not updated)&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;ul&gt;</li> <li>&lt;b&gt;Holiday Processing for Standing Instructions&lt;/b&gt;</li> <li>&lt;li&gt;The Holiday Processing can configured for the 07<sup>th</sup> day of execution 07<sup>th</sup> as defined by the business rules of the bank&lt;/li&gt;</li> <li>&lt;li&gt;If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>processed the day before the bank holiday&lt;/li&gt; &lt;li&gt;If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day. &lt;li&gt; &lt;li&gt;Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days. &lt;li&gt; &lt;ul&gt;</p> <p>&lt;li&gt;&lt;b&gt;Scenario 1: <math>\cap_{\gamma}^{\Downarrow}</math>Day Of Execution<math>\cap_{\gamma}^{\Downarrow}</math> is <math>\cap_{\gamma}^{\Downarrow}</math>Previous Day<math>\cap_{\gamma}^{\Downarrow}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System identifies that the next day is a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out till the next working day.&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;&lt;b&gt;Scenario 2: <math>\cap_{\gamma}^{\Downarrow}</math>Day Of Execution <math>\cap_{\gamma}^{\Downarrow}</math> is the <math>\cap_{\gamma}^{\Downarrow}</math>Next Working Date<math>\cap_{\gamma}^{\Downarrow}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System checks if the previous day was a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out from the last working date.&lt;/li&gt;&lt;/ol&gt;</p> <p>&lt;/ul&gt; &lt;br&gt;This method works as follows : &lt;br&gt; &lt;br&gt; generate transactionReferenceNumber. {@link UUIDGenerator#generate()} This method works as follows : &lt;br&gt; &lt;pre&gt; Event Id = STANDING_EVENT_CODE boolean isAllowed = true; List&lt;InstructionInstance&gt; instructionInstanceList; &amp;lt;i&amp;gt;&amp; First, it will check whether this instruction execution is allowed or not. If it is not allowed, then the SI is treated to be a failed instruction.&amp;lt;/i&amp;gt; try { Call {@link IDemandDepositAccount#validateStatus()}. }catch (Exception e) { &amp;lt;i&amp;gt;&amp; Log the message, and execute next SweepOut Instruction&amp;lt;/i&amp;gt; isAllowed = false; } &amp;lt;i&amp;gt;&amp; If the validate status allows the instruction to be processed, then the actual instruction execution is processed.&amp;lt;/i&amp;gt; if(isAllowed) { try { instructionInstanceList = call {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IDDASettlementAdapter#processSettlement() pass requestedSweepOutAmount. } catch(Exception e) {  <i>&amp;lt;i&amp;gt;</i>This may occur if the beneficiary account don't allow credit to occur.<i>&amp;lt;/i&amp;gt;</i> <i>&amp;lt;i&amp;gt;</i>In this case also, the instruction is treated as a failed instruction<i>&amp;lt;/i&amp;gt;</i> isAllowed = false; } } if instruction execution failed if(!isAllowed) { check if(retriesCount &amp;lt; maxRetries) { if(retriesCount == 0) actualExecutionDate = {@link RecurrencePatter#calcNexExecutionDate()} nextExecutionDate = next working day of the bank. retriesCount++; <i>&amp;lt;i&amp;gt;</i>Since a failed instruction is to be processed again, if maxRetries count permits.<i>&amp;lt;/i&amp;gt;</i> } else (retriesCount == maxRetries) { ignore failed instruction. reset retriesCount to 0. nextExecutionDate = actualExecutionDate. <i>&amp;lt;i&amp;gt;</i>If maxRetries count don't allow an instruction to be processed again, the instruction will be ignored.<i>&amp;lt;/i&amp;gt;</i> } } if(actualExecutionDate == nextExecutionDate) { <i>&amp;lt;i&amp;gt;</i>If an instruction execution due to its failure, and the actual next instruction execution falls on the same day, then the failed instruction will be ignored. Only one SI will be processed on a given day.<i>&amp;lt;/i&amp;gt;</i> reset retriesCount to 0. ignore failed instruction. } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate executionDate = pickupDate; subInstructionNumber++; Log Instraction instance for the execution of the instruction. } copy nextExecutionDate to lastExecutionDate. copy pickupDate to lastPickupDate calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> nextExecutionDate. update StandingInstruction by calling {@link StandingInstructionRepository#updateStandingInstru ction()} Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link StandingInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link StandingInstructionRequest#eventCode} Prepare {@link TransactionAmount} object from the standing instruction amount. {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link StandingInstructionRequest#transactionBranch} {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; @param instructionRequest </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	closeSettlementDetails	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	closeSettlementDetails	This method close the ComponentPreferenceDTO stored against the reference number and the accountId. @param componentPreferenceDto	This method is exposed to perform close operation for SweepOutInstruction @param sweepOutInstruction
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	This is the concrete implementation of the execute method. This method executes the Sweepout Instruction as follows: <ol style="list-style-type: none"> <li>The Sweep out is initiated during End Of Day Batch processing if the EpochType is set as EOD, if it is set up as BOD, it initiates in Beginning of the Operation Day.</li> <li>The system checks the accounts which are due for sweep out based on the Sweep out Start Date and the Recurrence pattern maintained.</li> <li>The System Validates end date for the instruction. The Conditions when the Sweep out is to be terminated is maintained in the recurrence pattern.</li> <li>System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account</li> <li>The system checks if the previous calendar date was a holiday, refer holiday processing explained in the documentation.</li> <li>System identifies the accounts eligible to execute Sweep outs on this processing date</li> <li>The System Checks the following balances defined for the sweep out arrangement. <ul style="list-style-type: none"> <li>The Cash Balance ( Minus the Hold)</li> <li>Minimum amount To be retained</li> <li>Threshold Balance</li> <li>Maximum Sweep Out amount</li> <li>The</li> </ul> </li> </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Amount Basis to arrive at the sweep out amount.</p> <ul style="list-style-type: none"> <li>The Execution of the instruction is based on the level and the priority defined for the accounts.</li> <li>The System Executes the transfers-Amounts which are greater than the maximum limit specified in the sweep out instruction are swept out to a pre-specified Savings account or a Time Deposit account.</li> <li>The System Sweeps out funds based on the Sweep Out Amount Basis Defined during Set up</li> <li>The system updates the book balance and passes the accounting entries (posting dated will be the current system date)</li> <li>The System maintains a Transaction Reference number for Audit purpose</li> <li>The System updates the Next Execution date . (For Accounts which the Sweep out Instructions ends the next execution date is not updated.</li> <li>The system executes the next instruction in the same way as the first instruction execution.</li> </ul> <p><b>Holiday Processing for Sweep Out Instructions</b></p> <ul style="list-style-type: none"> <li>The Operations Executive can configure the day of execution as defined by the business rules of the bank</li> <li>If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday</li> <li>If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day.</li> <li>Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days.</li> </ul> <p><b>Scenario 1: Day Of Execution is Previous Day</b></p> <ul style="list-style-type: none"> <li>The System identifies that the next day is a holiday in the Bank Calendar</li> <li>The System validates</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>all the account due for sweep out till the next working day.</p> <p><b>Scenario 2:</b> Day Of Execution is the Next Working Date</p> <p>The System checks if the previous day was a holiday in the Bank Calendar</p> <p>The System validates all the account due for sweep out from the last working date.</p> <p>For Example:</p> <ul style="list-style-type: none"> <li><b>Single provider Single beneficiary</b> <ul style="list-style-type: none"> <li>The Cash Balance ( Minus the Hold) = 25000</li> <li>Overdraft Balance (If considered) = NA</li> <li>Minimum amount To be retained = 5000</li> <li>Threshold Balance= NA</li> <li>Maximum Sweep Out amount = 15000</li> <li>Amount basis: Percentage (25%)</li> <li>The sweep out amount = (Cash Balance - Minimum amount to be retained)Amount basis</li> <li><math>(25000-5000)25\% = 5000</math></li> <li>Since 5000 is less than 15000 (The Max Sweep out Amount ) the amount swept out is 5000.00</li> </ul> </li> <li><b>Single Provider Multiple Beneficiary</b> <ul style="list-style-type: none"> <li>The Cash Balance ( Minus the Hold) = 25000</li> <li>Overdraft Balance (If considered) = NA</li> <li>Minimum amount To be retained = 5000</li> <li>Threshold Balance= NA</li> <li>Maximum Sweep Out amount = 15000</li> <li>Amount basis: Multiples of 400</li> <li>Beneficiary A 20%</li> <li>Beneficiary B 50%</li> <li>Beneficiary C 30%</li> </ul> </li> </ul> <p>The Sweep Out Amount = <math>(25000-5000) = 15000</math></p> <p>Since the amount basis is multiples of 400 only 14800 (40037 times) is eligible to be swept out</p> <ul style="list-style-type: none"> <li>Beneficiary A = <math>14800 \times 20\% = 2960</math></li> <li>Beneficiary B = <math>14800 \times 50\% = 7400</math></li> <li>Beneficiary C = <math>14800 \times 30\% = 4440</math></li> </ul> <p>If any sweep out instruction fails the amount will</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>remain with the provider account. For instance the sweep out to Beneficiary B fails, then only 7440 ( to A- 2960 + to C 4440) is debited from the provider account.</p> <p>This method works as follows :</p> <pre> Event Id = SWEEP_OUT_EVENT_CODE boolean isAllowed = true; try { Call IDemandDepositAccount#validateStatus(). } catch (Exception e) { Log the message, and execute next SweepOut Instruction isAllowed = false; } eligibleSweepOutAmount = SweepOutInstruction#deriveEligibleSweepOutAmount() generate transactionReferenceNumber UUIDGenerator#generate() List&lt;InstructionInstance&gt; instructionInstanceList; for( BeneficiaryAccount beneficiaryAccount : SweepOutInstruction#getBeneficiaryAccountList()) { requestedSweepOutAmount = SweepOutInstruction#deriveRequestedSweepOutAmount(eligibleSweepOutAmount) if(requestedSweepOutAmount &lt;= eligibleSweepOutAmount) { if(isAllowed) { instructionInstanceList = call IDDASettlementAdapter#processSettlement() pass requestedSweepOutAmount. } } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDueDate = nextExecutionDate executionDate = pickupDate; subInstructionNumber++; Log Instruction instance for the execution of the instruction. } calculate new nextExecutionDate(); call derivePickupDate() to derive new pickupDate on the basis of new nextExecutionDate. update SweepOutInstruction by </pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> calling {@link SweepOutInstructionRepository#updateSweepOutInst ruction()} &amp;lt;i&amp;gt;Fee will be charged on every sweep out instruction execution, ie. irrespective of the sub instructions, fee management service will be called only once.&amp;lt;/i&amp;gt; Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link SweepOutInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link SweepOutInstructionRequest#eventCode} {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link SweepOutInstructionRequest#transactionBranch} TODO {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; @param instructionRequest </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	This method executes the component preference online.this will credit the settlement GL with necessary amount.If it throws the exception then corresponding executeDebitInstruction should catch it and mark the instruction as failed. @param componentPreferenceDto @param accountId @param debitAmount @param valueDate TODO @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	This method is invoked for Final Settlement Processing. NOTE: Account Closure Request has already been raised. @param sessionContext @param accountId @param applicationContext @param closureRequestID @param debitResponse @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	This method is invoked for Final Settlement Processing. NOTE: Account Closure Request is not being raised previously. @param sessionContext @param accountId @param closureDTO @param applicationContext @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	This method ensures whether the any unsettled amount is present or not. If any unsettled amount is found, Account Closure process is stopped. @param accountId @param settlementResponse
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	Execution of Periodic Funds Transfer instructions (Standing Instruction) is handled at batch process, Beginning of Day operations (BOD). The execution of Periodic Funds Transfer instructions happens on the previous working day, or the next working day in case the execution day is a holiday, This is the concrete implementation of the execute method. This method executes the Standing Instruction as follows: <ol><li>The Periodic Transfer Instruction will always be initiated during BOD.</li> <li>The system checks the accounts which are due for Periodic Transfer based on the Periodic Transfer Start Date and the Recurrence pattern maintained.</li> <li>The System Validates end date for the instruction. The Conditions when the Periodic Transfer Instruction is to be terminated is

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>maintained in the recurrence pattern.</p> <ul style="list-style-type: none"> <li>System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account</li> <li>The system validates the holiday processing rules (Refer Holiday Processing)</li> <li>System identifies the accounts eligible to execute Periodic Transfer Instruction on this processing date</li> <li>The System Checks If the Funds are available in the Provider accounts</li> <li>The Execution of the instruction is based on the priority defined for the accounts.</li> <li>The System Executes the transfers</li> <li>The system updates the book balance and passes the accounting entries (posting dated will be the current system date)</li> <li>The System maintains a Transaction Reference number for Audit purpose</li> <li>The System updates the Next Execution date. (For Accounts, which the Periodic Transfer Instruction ends the next execution date is not updated)</li> </ul> <p><b>Holiday Processing for Standing Instructions</b></p> <ul style="list-style-type: none"> <li>The Holiday Processing can configured for the <math>\cap_{\gamma}^{\Downarrow}</math> day of execution <math>\cap_{\gamma}^{\Downarrow}</math> as defined by the business rules of the bank</li> <li>If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday</li> <li>If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day.</li> <li>Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days.</li> </ul> <p><b>Scenario 1: <math>\cap_{\gamma}^{\Downarrow}</math> Day Of Execution <math>\cap_{\gamma}^{\Downarrow}</math> is <math>\cap_{\gamma}^{\Downarrow}</math> Previous Day <math>\cap_{\gamma}^{\Downarrow}</math></b></p> <ul style="list-style-type: none"> <li>The System</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>identifies that the next day is a holiday in the Bank Calendar</p> <ul style="list-style-type: none"> <li>The System validates all the account due for sweep out till the next working day.</li> </ul> <p><b>Scenario 2: <math>\cap</math> Day Of Execution <math>\cap</math> is the <math>\cap</math> Next Working Date</b></p> <ul style="list-style-type: none"> <li>The System checks if the previous day was a holiday in the Bank Calendar</li> <li>The System validates all the account due for sweep out from the last working date.</li> </ul> <p>This method works as follows :</p> <pre> generate transactionReferenceNumber. @link UUIDGenerator#generate() This method works as follows : <pre> Event Id = STANDING_EVENT_CODE boolean isAllowed = true; List&lt;InstructionInstance&gt; instructionInstanceList; First, it will check whether this instruction execution is allowed or not. If it is not allowed, then the SI is treated to be a failed instruction. try { Call @link IDemandDepositAccount#validateStatus(). } catch (Exception e) { Log the message, and execute next SweepOut Instruction; isAllowed = false; } If the validate status allows the instruction to be processed, then the actual instruction execution is processed. if(isAllowed) { try { instructionInstanceList = call @link IDDASettlementAdapter#processSettlement() pass requestedSweepOutAmount. } catch(Exception e) { This may occur if the beneficiary account don't allow credit to occur. In this case also, the instruction is treated as a failed instruction; } } if instruction execution failed if(!isAllowed) { check if(retriesCount &lt; maxRetries) if(retriesCount == 0) actualExecutionDate = @link </pre> </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> RecurrencePatter#calcNexExecutionDate() nextExecutionDate = next working day of the bank. retriesCount++; &amp;lt;i&amp;gt;Since a failed instruction is to be processed again, if maxRetries count permits.&amp;lt;/i&amp;gt; } else (retriesCount == maxRetries) { ignore failed instruction. reset retriesCount to 0. nextExecutionDate = actualExecutionDate. &amp;lt;i&amp;gt;If maxRetries count don't allow an instruction to be processed again, the instruction will be ignored.&amp;lt;/i&amp;gt; } } if(actualExecutionDate == nextExecutionDate) { &amp;lt;i&amp;gt;If an instruction execution due to its failure, and the actual next instruction execution falls on the same day, then the failed instruction will be ignored. Only one SI will be processed on a given day.&amp;lt;/i&amp;gt; reset retriesCount to 0. ignore failed instruction. } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate executionDate = pickupDate; subInstructionNumber++; Log Instraction instance for the execution of the instruction. } copy nextExecutionDate to lastExecutionDate. copy pickupDate to lastPickupDate calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update StandingInstruction by calling {@link StandingInstructionRepository#updateStandingInstru ction()} Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link StandingInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> StandingInstructionRequest#eventCode} Prepare {@link TransactionAmount} object from the standing instruction amount. {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link StandingInstructionRequest#transactionBranch} {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO: Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; </pre>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlement	Refer {@link DDASettlementAdapter#processSettlement} @param accountId @param settlementReferenceNumber @param amountForSettlement @param forceDebitFlag @param eventID @param narration TODO @param valueDate TODO @param userReferenceNum @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	processSettlementForBill	@param sourcingResponse @param instructionRequest @param debitAccountingEventAmountTagDetailsDTO @return @throws FatalException	Execution of Periodic Funds Transfer instructions (Standing Instruction) is handled at batch process, Beginning of Day operations (BOD). The execution of Periodic Funds Transfer instructions happens on the previous working day, or the next working day in case the execution day is a holiday, This is the concrete implementation of the execute method. This method executes the Standing Instruction as follows: <ul style="list-style-type: none"> <li>&lt;li&gt;The Periodic Transfer Instruction will always be initiated during BOD.&lt;/li&gt;</li> <li>&lt;li&gt;The system checks the accounts which are due for Periodic Transfer based on the Periodic Transfer Start Date and the Recurrence pattern maintained.&lt;/li&gt;</li> <li>&lt;li&gt;The System Validates end date for the instruction. The Conditions when the Periodic Transfer Instruction is to be terminated is maintained in the recurrence pattern.&lt;/li&gt;</li> <li>&lt;li&gt;System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account&lt;/li&gt;</li> <li>&lt;li&gt;The system validates the holiday processing rules (Refer Holiday Processing)&lt;/li&gt;</li> <li>&lt;li&gt;System identifies the accounts eligible to execute Periodic Transfer Instruction on this processing date&lt;/li&gt;</li> <li>&lt;li&gt;The System Checks If the Funds are available in the Provider accounts&lt;/li&gt;</li> <li>&lt;li&gt;The Execution of the instruction is based on the priority defined for the accounts.&lt;/li&gt;</li> <li>&lt;li&gt;The System Executes the transfers&lt;/li&gt;</li> <li>&lt;li&gt;The system updates the book balance and passes the accounting entries (posting dated will be the current system date)&lt;/li&gt;</li> <li>&lt;li&gt;The System maintains a Transaction Reference number for Audit purpose&lt;/li&gt;</li> <li>&lt;li&gt;The System updates the Next Execution date. (For Accounts, which the Periodic Transfer Instruction ends the next execution date is not updated)&lt;/li&gt;</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Processing for Standing Instructions&lt;/b&gt; &lt;li&gt;The Holiday Processing can configured for the <math>\cap_{\gamma}^{\text{H}}</math> day of execution<math>\cap_{\gamma}^{\text{H}}</math> as defined by the business rules of the bank&lt;/li&gt; &lt;li&gt;If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday&lt;/li&gt; &lt;li&gt;If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day. &lt;li&gt;&lt;li&gt;Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days. &lt;li&gt; &lt;ul&gt;</p> <p>&lt;li&gt;&lt;b&gt;Scenario 1: <math>\cap_{\gamma}^{\text{H}}</math> Day Of Execution<math>\cap_{\gamma}^{\text{H}}</math> is <math>\cap_{\gamma}^{\text{H}}</math> Previous Day<math>\cap_{\gamma}^{\text{H}}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System identifies that the next day is a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out till the next working day.&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;&lt;b&gt;Scenario 2: <math>\cap_{\gamma}^{\text{H}}</math> Day Of Execution <math>\cap_{\gamma}^{\text{H}}</math> is the <math>\cap_{\gamma}^{\text{H}}</math> Next Working Date<math>\cap_{\gamma}^{\text{H}}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System checks if the previous day was a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out from the last working date.&lt;/li&gt;&lt;/ol&gt;</p> <p>&lt;/ul&gt; &lt;br&gt;This method works as follows : &lt;br&gt; &lt;br&gt; generate transactionReferenceNumber. {<a href="#">@link UUIDGenerator#generate()</a>} This method works as follows : &lt;br&gt; &lt;pre&gt; Event Id = STANDING_EVENT_CODE boolean isAllowed = true; List&lt;InstructionInstance&gt; instructionInstanceList; &amp;lt;i&gt;&amp;gt;First, it will check whether this instruction execution is allowed or not. If it is not allowed, then the SI is treated to be a failed instruction.&amp;lt;/i&gt;&amp;gt; try { <a href="#">Call {@link IDemandDepositAccount#validateStatus()</a>. }catch</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> (Exception e) {     &lt;i&gt;Log the message, and execute next SweepOut Instruction&lt;/i&gt;     isAllowed = false; } &lt;i&gt;If the validate status allows the instruction to be processed, then the actual instruction execution is processed.&lt;/i&gt; if(isAllowed) {     try {         instructionInstanceList = call {             @link IDDASettlementAdapter#processSettlement()         } pass requestedSweepOutAmount.     } catch(Exception e) {         &lt;i&gt;This may occur if the beneficiary account don't allow credit to occur.&lt;/i&gt;         &lt;i&gt;In this case also, the instruction is treated as a failed instruction&lt;/i&gt;         isAllowed = false;     }     if instruction execution failed     if(!isAllowed) {         check if(retriesCount &lt; maxRetries)         if(retriesCount == 0)             actualExecutionDate = {                 @link RecurrencePatter#calcNexExecutionDate()             }             nextExecutionDate = next working day of the bank.             retriesCount++;         &lt;i&gt;Since a failed instruction is to be processed again, if maxRetries count permits.&lt;/i&gt;     } else (retriesCount == maxRetries)     {         ignore failed instruction.         reset retriesCount to 0.         nextExecutionDate = actualExecutionDate.         &lt;i&gt;If maxRetries count don't allow an instruction to be processed again, the instruction will be ignored.&lt;/i&gt;     }     if(actualExecutionDate == nextExecutionDate) {         &lt;i&gt;If an instruction execution due to its failure, and the actual next instruction execution falls on the same day, then the failed instruction will be ignored. Only one SI will be processed on a given day.&lt;/i&gt;         reset retriesCount to 0.         ignore failed instruction.     }     subInstructionNumber = 0;     for(InstructionInstance instructionInstance : instructionInstanceList) {         set = executionDue Date = nextExecutionDate         executionDate = pickupDate; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>subInstructionNumber++; Log Instruction instance for the execution of the instruction. } copy nextExecutionDate to lastExecutionDate. copy pickupDate to lastPickupDate calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update StandingInstruction by calling {@link StandingInstructionRepository#updateStandingInstruction()} Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link StandingInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link StandingInstructionRequest#eventCode} Prepare {@link TransactionAmount} object from the standing instruction amount. {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link StandingInstructionRequest#transactionBranch} {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} </pre> @param instructionRequest
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deriveAccountForReverseSweepOut	This method will derive the target balance for Sweep out instruction execution. @param accountId @param settlementReferenceNumber @return	This method will calculate the processing amount on the basis of the input eligible amount. This method will return the processing amount on the basis of the following:-  @param eligibleAmountForCalculationFromProvider @return Money
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	deriveAccountForReverseSweepOut	This method will derive the target balance for Sweep out instruction execution. @param accountId @param settlementReferenceNumber @return	
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	This method executes the component preference online.this will credit the settlement GL with necessary amount.If it throws the exception then corresponding executeDebitInstruction should catch it and mark the instruction as failed. @param componentPreferenceDto @param accountId @param debitAmount @param valueDate TODO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	This method is invoked for Final Settlement Processing. NOTE: Account Closure Request is not being raised previously. @param sessionContext @param accountId @param closureDTO @param applicationContext @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	@see com.ofss.fc.app.dda.service.core.IDemandDepositTransactionService#issueMCToSelf(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String, com.ofss.fc.datatype.dto.MoneyDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> }  This method is used in case the component preference does not exist. The settlement service in this case  will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> }  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> }  This method is used in case the component preference does not exist. The settlement service in this case  will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> }   @param accountId @param amountForSettlement @param eventId @param serviceCode @param instruction @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.  
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method executes the component preference online.this will credit the settlement GL with necessary amount.If it throws the exception then corresponding executeDebitInstruction should catch it and mark the instruction as failed. @param componentPreferenceDto @param accountId @param debitAmount @param valueDate TODO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method is invoked for Final Settlement Processing. NOTE: Account Closure Request has already been raised. @param sessionContext @param accountId @param applicationContext @param closureRequestID @param debitResponse @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method is invoked for Final Settlement Processing. NOTE: Account Closure Request is not being raised previously. @param sessionContext @param accountId @param closureDTO @param applicationContext @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	@see com.ofss.fc.app.dda.service.core.IDemandDepositTransactionService#issueMCToSelf(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String, com.ofss.fc.datatype.dto.MoneyDTO)
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	Method used for miscellaneous credits. Miscellaneous credits are mostly for patch works. <p>The method is also used for salary credit through file upload. The employer account is debited in the file processing header and amount will be banked in an ledger. Then this method will be called to credit the employee account, debiting for the same ledger. </p>The method will call settlement to settle the credit amount. If the bsb belongs to an external branch, then settlement mode will be sent as payments. For internal accounts, the bsb must be sent as null.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	Method used for miscellaneous debits. Miscellaneous debits are mostly for patch works. <p>The method is also used for salary credit through file upload. The employer account is debited in the file processing header and amount will be banked in an ledger. This method will be called to perform this action. </p>The method will call settlement to settle the credit amount. If the bsb belongs to an external branch, then settlement mode will be sent as payments. For internal accounts, the bsb must be sent as null.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.   @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}   @param accountId @param amountForSettlement @param eventId @param serviceCode @param instruction @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param instruction @param @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreference	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference exist The settlement service in this case   will execute the preference.   @param amountForSettlement @param nsfTreatment @param narration TODO @param instruction @param @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the   settlement service {@link SettlementExecutionService#executePreferenceOnline}  This method is used in case the component preference does not exist. The settlement service in this case   will create a component preference and then execute the preference.  

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchStalePeriodForInstrument	This method calls {@link InstrumentCategoryApplicationService#fetchInstrumentStalePeriod} This method will return the StaleCheckInquiryResponse. @param instrumentDate @return StaleCheckInquiryResponse	<ul style="list-style-type: none"> <li>Cheque can be used to withdraw cash in the account. If withdraw is happening using the cheque then the same is validated. Instrument(Cheque) is validated by calling {@link ChequeManagementService#fetchCheque(cheque: Cheque): Cheque}.</li> <li>Do not allow transaction if cheque status is marked as paid, stopped or returned.</li> <li>Cheque can be used to withdraw cash in the account. If withdraw is happening using the cheque then the instrument date will be validated. Instrument date is validated if InstrumentDetail.instrumentNumber is not null.</li> <li>Cheque date will be validated by cheque validity period defined at clearing module.</li> <li>System will not allow transaction if validity period expires. E.g. suppose cheque validity period is of six months and date mention on cheque is 10th March, 2009 and then validity period expires on 9th Sep, 2009.</li> </ul> <p>The validation is done as follows :</p> <ul style="list-style-type: none"> <li>Get instance of the IDDASettlementAdapter using DDASettlementAdapterFactory</li> <li>Call {@link IDDASettlementAdapter#fetchStalePeriodForInstrument(CategoryId)}. It returns Stale Period as output.</li> <li>@param accountTransaction</li> </ul>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchStalePeriodForInstrument	This method calls {@link InstrumentCategoryApplicationService#fetchInstrumentStalePeriod} This method will return the StaleCheckInquiryResponse. @param instrumentDate @return StaleCheckInquiryResponse	This method is used to get the Stale period for an instrument. {@link com.ofss.fc.app.pc.service.instrument.InstrumentCategoryApplicationService.checkIfInstrumentIsStale(SessionContext, InstrumentType, Date, Integer)} @param instrumentDate @return StaleCheckInquiryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchTDPayoutLinkageDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchTDPayoutLinkageDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method will be used inquire the linkage details of the account with another CASA, TD or Loan account. The method works as follows : <ul style="list-style-type: none"> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForFees} to fetch the list of parent accounts for which the CASA acts as an alternate account for fees posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForDebitInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for debit interest posting.</li> <li>Call the method {@link AlternateAccountService#fetchParentAccountsForCreditInterest} to fetch the list of parent accounts for which the CASA acts as an alternate account for credit interest posting.</li> <li>Call the method {@link IDDALoanAdapter#checkOffsetAccountLinkage} to get the list of Loan accounts for which the CASA acts as an Offset account.</li> <li>Call the method {@link IDDALoanAdapter#checkLoanAccountPayInstruction} to get the list of Loan accounts for which the CASA acts as an installment payment account.</li> <li>Call the method {@link } to get the list of credit card numbers for which the CASA acts as a payment account.</li> <li>Call the method {@link IDDASettlementAdapter#fetchTDPayoutLinkageDetail</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IsForDDAccount } to get the list of Term Deposit accounts and Deposit numbers for which the CASA acts as payout account for Principal or Interest component.</p> <ul style="list-style-type: none"> <li>Combine the response received by each of the methods in a single collection along with the linkage type and return.</li> </ul> <p>@param accountId</p>
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferences	@param settlementId @param accountId @return	<p>This is the concrete implementation of the execute method. This method executes the Sweepout Instruction as follows:</p> <ul style="list-style-type: none"> <li>The Sweep out is initiated during End Of Day Batch processing if the EpochType is set as EOD, if it is set up as BOD, it initiates in Beginning of the Operation Day.</li> <li>The system checks the accounts which are due for sweep out based on the Sweep out Start Date and the Recurrence pattern maintained.</li> <li>The System Validates end date for the instruction. The Conditions when the Sweep out is to be terminated is maintained in the recurrence pattern.</li> <li>System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account</li> <li>The system checks if the previous calendar date was a holiday, refer holiday processing explained in the documentation.</li> <li>System identifies the accounts eligible to execute Sweep outs on this processing date</li> <li>The System Checks the following balances defined for the sweep out arrangement: <ul style="list-style-type: none"> <li>The Cash Balance ( Minus the Hold)</li> <li>Minimum amount To be retained</li> <li>Threshold Balance</li> <li>Maximum Sweep Out amount</li> <li>The Amount Basis to arrive at the sweep out amount.</li> </ul> </li> </ul> <p>The Execution of the instruction is based on</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the <code>level</code> and the priority defined for the accounts.</p> <ul style="list-style-type: none"> <li>The System Executes the transfers- Amounts which are greater than the maximum limit specified in the sweep out instruction are swept out to a pre-specified Savings account or a Time Deposit account.</li> <li>The System Sweeps out funds based on the Sweep Out Amount Basis Defined during Set up</li> <li>The system updates the book balance and passes the accounting entries (posting dated will be the current system date)</li> <li>The System maintains a Transaction Reference number for Audit purpose</li> <li>The System updates the Next Execution date . (For Accounts which the Sweep out Instructions ends the next execution date is not updated.</li> <li>The system executes the next instruction in the same way as the first instruction execution.</li> </ul> <ul style="list-style-type: none"> <li><b>Holiday Processing for Sweep Out Instructions</b></li> <li>The Operations Executive can configure the day of execution as defined by the business rules of the bank</li> <li>If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday</li> <li>If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day.</li> <li>Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days.</li> </ul> <ul style="list-style-type: none"> <li><b>Scenario 1: Day Of Execution is Previous Day</b></li> <li>The System identifies that the next day is a holiday in the Bank Calendar</li> <li>The System validates all the account due for sweep out till the next working day.</li> </ul> <ul style="list-style-type: none"> <li><b>Scenario 2: Day</b></li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Of Execution <math>\cap_{\gamma}^{\downarrow}</math> is the <math>\cap_{\gamma}^{\downarrow}</math> Next Working Date <math>\cap_{\gamma}^{\downarrow}</math></p> <ul style="list-style-type: none"> <li>&lt;li&gt;The System checks if the previous day was a holiday in the Bank Calendar&lt;/li&gt;</li> <li>&lt;li&gt;The System validates all the account due for sweep out from the last working date.&lt;/li&gt;&lt;/li&gt;</li></ul> <p>For Example:</p> <ul style="list-style-type: none"> <li>&lt;li&gt;&lt;b&gt;Single provider Single beneficiary&lt;/b&gt;</li> <li>&lt;li&gt;The Cash Balance ( Minus the Hold) = 25000&lt;/li&gt;</li> <li>&lt;li&gt;Overdraft Balance (If considered) = NA&lt;/li&gt;</li> <li>&lt;li&gt;Minimum amount To be retained = 5000&lt;/li&gt;</li> <li>&lt;li&gt;Threshold Balance= NA&lt;/li&gt;</li> <li>&lt;li&gt;Maximum Sweep Out amount = 15000&lt;/li&gt;</li> <li>&lt;li&gt;Amount basis: Percentage (25%)&lt;/li&gt;</li> <li>&lt;li&gt;The sweep out amount = (Cash Balance <math>\cap_{\gamma}^{\downarrow}</math> Minimum amount to be retained)Amount basis&lt;/li&gt;</li> <li>&lt;li&gt;=(25000-5000)25% =5000&lt;/li&gt;</li> <li>&lt;li&gt;Since 5000 is less than 15000 (The Max Sweep out Amount ) the amount swept out is 5000.00&lt;/li&gt;&lt;/li&gt;</li></ul> <ul style="list-style-type: none"> <li>&lt;li&gt;&lt;b&gt;Single Provider Multiple Beneficiary&lt;/b&gt;</li> <li>&lt;li&gt;The Cash Balance ( Minus the Hold) = 25000&lt;/li&gt;</li> <li>&lt;li&gt;Overdraft Balance (If considered) = NA&lt;/li&gt;</li> <li>&lt;li&gt;Minimum amount To be retained = 5000&lt;/li&gt;</li> <li>&lt;li&gt;Threshold Balance= NA&lt;/li&gt;</li> <li>&lt;li&gt;Maximum Sweep Out amount = 15000&lt;/li&gt;</li> <li>&lt;li&gt;Amount basis: Multiples of 400&lt;/li&gt;</li></ul> <ul style="list-style-type: none"> <li>&lt;li&gt;Beneficiary A <math>\cap_{\gamma}^{\downarrow}</math> 20%&lt;/li&gt;</li> <li>&lt;li&gt;Beneficiary B <math>\cap_{\gamma}^{\downarrow}</math> 50%&lt;/li&gt;</li> <li>&lt;li&gt;Beneficiary C <math>\cap_{\gamma}^{\downarrow}</math> 30%&lt;/li&gt;</li></ul> <p>&lt;li&gt;The Sweep Out Amount = (25000-5000) = 15000&lt;/li&gt;</p> <p>&lt;li&gt;Since the amount basis is multiples of 400 only 14800 (40037 times) is eligible to be swept out&lt;/li&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Beneficiary A = 1480020% = 2960&lt;/li&gt;</li> <li>&lt;li&gt;Beneficiary B = 14800 50%= 7400&lt;/li&gt;</li> <li>&lt;li&gt;Beneficiary C = 14800 30% = 4440&lt;/li&gt;</li></ul> <p>&lt;li&gt;If any sweep out instruction fails the amount will remain with the provider account. For instance the sweep out to <math>\cap_{\gamma}^{\downarrow}</math> Beneficiary B <math>\cap_{\gamma}^{\downarrow}</math> Fails, then only</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>7440 ( to A- 2960 + to C 4440) is debited from the provider account.</p> <p>This method works as follows :</p> <pre> Event Id = SWEEP_OUT_EVENT_CODE boolean isAllowed = true; try { Call IDemandDepositAccount#validateStatus(). }catch (Exception e) { &amp;lt;i&amp;gt;Log the message, and execute next SweepOut Instruction&amp;lt;/i&amp;gt; isAllowed = false; } eligibleSweepOutAmount = {@link SweepOutInstruction#deriveEligibleSweepOutAmount() } generate transactionReferenceNumber {@link UUIDGenerator#generate() } List&amp;lt;InstructionInstance&amp;gt; instructionInstanceList; for( {@link BeneficiaryAccount} beneficiaryAccount : {@link SweepOutInstruction#getBeneficiaryAccountList()}) { requestedSweepOutAmount = {@link SweepOutInstruction#deriveRequestedSweepOutAmount( eligibleSweepOutAmount) } if(requestedSweepOutAmount &amp;lt;= eligibleSweepOutAmount) { if(isAllowed) { instructionInstanceList = call {@link IDDASettlementAdapter#processSettlement() } pass requestedSweepOutAmount. } } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate executionDate = pickupDate; subInstructionNumber++; Log Instruction instance for the execution of the instruction. } calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update SweepOutInstruction by calling {@link SweepOutInstructionRepository#updateSweepOutInst </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> ruction() &amp;lt;i&amp;gt;Fee will be charged on every sweep out instruction execution, ie. irrespective of the sub instructions, fee management service will be called only once.&amp;lt;/i&amp;gt; Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link SweepOutInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link SweepOutInstructionRequest#eventCode} {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link SweepOutInstructionRequest#transactionBranch} TODO {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} &lt;/pre&gt; @param instructionRequest </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromReferences	@param settlementId @param accountId @return	<p>Execution of Periodic Funds Transfer instructions (Standing Instruction) is handled at batch process, Beginning of Day operations (BOD). The execution of Periodic Funds Transfer instructions happens on the previous working day, or the next working day in case the execution day is a holiday, This is the concrete implementation of the execute method. This method executes the Standing Instruction as follows: &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;The Periodic Transfer Instruction will always be initiated during BOD.&lt;/li&gt;</li> <li>&lt;li&gt;The system checks the accounts which are due for Periodic Transfer based on the Periodic Transfer Start Date and the Recurrence pattern maintained.&lt;/li&gt;</li> <li>&lt;li&gt;The System Validates end date for the instruction. The Conditions when the Periodic Transfer Instruction is to be terminated is maintained in the recurrence pattern.&lt;/li&gt;</li> <li>&lt;li&gt;System checks the status of the provider and beneficiary Accounts. Business rules to be built to allow or reject based on the status of the account&lt;/li&gt;</li> <li>&lt;li&gt;The system validates the holiday processing rules (Refer Holiday Processing)&lt;/li&gt;</li> <li>&lt;li&gt;System identifies the accounts eligible to execute Periodic Transfer Instruction on this processing date&lt;/li&gt;</li> <li>&lt;li&gt;The System Checks If the Funds are available in the Provider accounts&lt;/li&gt;</li> <li>&lt;li&gt;The Execution of the instruction is based on the priority defined for the accounts.&lt;/li&gt;</li> <li>&lt;li&gt;The System Executes the transfers&lt;/li&gt;</li> <li>&lt;li&gt;The system updates the book balance and passes the accounting entries (posting dated will be the current system date)&lt;/li&gt;</li> <li>&lt;li&gt;The System maintains a Transaction Reference number for Audit purpose&lt;/li&gt;</li> <li>&lt;li&gt;The System updates the Next Execution date. (For Accounts, which the Periodic Transfer Instruction ends the next execution date is not updated)&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt; &lt;ul&gt; &lt;b&gt;Holiday</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Processing for Standing Instructions&lt;/b&gt; &lt;li&gt;The Holiday Processing can configured for the <math>\cap_{\gamma}^{\mathbb{J}}</math> day of execution<math>\cap_{\gamma}^{\mathbb{J}}</math> as defined by the business rules of the bank&lt;/li&gt; &lt;li&gt;If the Business rule indicates that the execution is to be processed the previous day, then the accounts for which the funds transfer is due will be processed the day before the bank holiday&lt;/li&gt; &lt;li&gt;If the date for execution is not configured to previous, the system will initiate the instruction order on the next working day. &lt;li&gt;&lt;li&gt;Transactions falling on a holiday will be executed on next working day, but if the frequency is daily, the transactions will only be executed on working days. &lt;li&gt; &lt;ul&gt;</p> <p>&lt;li&gt;&lt;b&gt;Scenario 1: <math>\cap_{\gamma}^{\mathbb{J}}</math> Day Of Execution<math>\cap_{\gamma}^{\mathbb{J}}</math> is <math>\cap_{\gamma}^{\mathbb{J}}</math> Previous Day<math>\cap_{\gamma}^{\mathbb{J}}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System identifies that the next day is a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out till the next working day.&lt;/li&gt; &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;&lt;b&gt;Scenario 2: <math>\cap_{\gamma}^{\mathbb{J}}</math> Day Of Execution <math>\cap_{\gamma}^{\mathbb{J}}</math> is the <math>\cap_{\gamma}^{\mathbb{J}}</math> Next Working Date<math>\cap_{\gamma}^{\mathbb{J}}</math>&lt;/b&gt; &lt;ol&gt; &lt;li&gt;The System checks if the previous day was a holiday in the Bank Calendar&lt;/li&gt; &lt;li&gt;The System validates all the account due for sweep out from the last working date.&lt;/li&gt;&lt;/ol&gt;</p> <p>&lt;/ul&gt; &lt;br&gt;This method works as follows : &lt;br&gt; &lt;br&gt; generate transactionReferenceNumber. {@link UUIDGenerator#generate()} This method works as follows : &lt;br&gt; &lt;pre&gt; Event Id = STANDING_EVENT_CODE boolean isAllowed = true; List&lt;InstructionInstance&gt; instructionInstanceList; &amp;lt;i&gt;&amp;gt;First, it will check whether this instruction execution is allowed or not. If it is not allowed, then the SI is treated to be a failed instruction.&amp;lt;/i&gt;&amp;gt; try { Call {@link IDemandDepositAccount#validateStatus()}. }catch</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> (Exception e) { &amp;lt;i&amp;gt; Log the message, and execute next SweepOut Instruction&amp;lt;/i&amp;gt; isAllowed = false; } &amp;lt;i&amp;gt;If the validate status allows the instruction to be processed, then the actual instruction execution is processed.&amp;lt;/i&amp;gt; if(isAllowed) { try { instructionInstanceList = call {@link IDDASettlementAdapter#processSettlement()} pass requestedSweepOutAmount. } catch(Exception e) { &amp;lt;i&amp;gt;This may occur if the beneficiary account don't allow credit to occur.&amp;lt;/i&amp;gt; &amp;lt;i&amp;gt;In this case also, the instruction is treated as a failed instruction&amp;lt;/i&amp;gt; isAllowed = false; } } if instruction execution failed if(!isAllowed) { check if(retriesCount &amp;lt; maxRetries) { if(retriesCount == 0) actualExecutionDate = {@link RecurrencePatter#calcNexExecutionDate()} nextExecutionDate = next working day of the bank. retriesCount++; &amp;lt;i&amp;gt;Since a failed instruction is to be processed again, if maxRetries count permits.&amp;lt;/i&amp;gt; } else (retriesCount == maxRetries) { ignore failed instruction. reset retriesCount to 0. nextExecutionDate = actualExecutionDate. &amp;lt;i&amp;gt;If maxRetries count don't allow an instruction to be processed again, the instruction will be ignored.&amp;lt;/i&amp;gt; } } if(actualExecutionDate == nextExecutionDate) { &amp;lt;i&amp;gt;If an instruction execution due to its failure, and the actual next instruction execution falls on the same day, then the failed instruction will be ignored. Only one SI will be processed on a given day.&amp;lt;/i&amp;gt; reset retriesCount to 0. ignore failed instruction. } subInstructionNumber = 0; for(InstructionInstance instructionInstance : instructionInstanceList) { set = executionDue Date = nextExecutionDate executionDate = pickupDate; </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>subInstructionNumber++; Log Instruction instance for the execution of the instruction. } copy nextExecutionDate to lastExecutionDate. copy pickupDate to lastPickupDate calculate new nextExecutionDate(); call derivePickupDate() to derive new picupdate on the basis of new nextExecutionDate. update StandingInstruction by calling {@link StandingInstructionRepository#updateStandingInstruction()} Prepare {@link DDAFeeRequestInfo} {@link DDAFeeRequestInfo#partyId} = {@link StandingInstructionRequest#partyId} {@link DDAFeeRequestInfo#accountId} = {@link InstructionKey#accountId} {@link DDAFeeRequestInfo#eventCode} = {@link StandingInstructionRequest#eventCode} Prepare {@link TransactionAmount} object from the standing instruction amount. {@link DDAFeeRequestInfo#transactionAmount} = transactionAmount {@link DDAFeeRequestInfo#transactionReferenceNumber} = transactionReferenceNumber {@link DDAFeeRequestInfo#currencyCodes} = Provider account CurrencyCode in String[] {@link DDAFeeRequestInfo#transactionBranch} = {@link StandingInstructionRequest#transactionBranch} {@link DDAFeeRequestInfo#numberOfPrintedPages} = null {@link DDAFeeRequestInfo#FeeAmounts} = null {@link DDAFeeRequestInfo#pricingTicketNumber} = null Call {@link FeeManagementService#checkAndLevyFees()} accounting entries will be passed as of nextExecutionDate before computing new ExecutionDate. pickupDate is posting date</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				nextExecutionDate is value date. TODO : Prepare AccountingTransactionEvent object Call {@link IDomainAccountingAdapter#raiseAccountingEvent()} </pre> @param instructionRequest
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferences	@param settlementId @param accountId @return	Method will get the comma separated string of all the account id in the component preferences.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferences	@param settlementId @param accountId @return	Method will get the comma separated string of all the account id in the component preferences.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferences	@param settlementId @param accountId @return	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferences	@param settlementId @param accountId @return	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getAllAccountsFromPreferencesForOnline	Method will get the comma separated string of all the account id in the component preferences.	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getBankNamesFromPreferences	Method will get the comma separated string of all the account id in the component preferences.	Method will get the comma separated string of all the account id in the component preferences.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	dapter			
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getBankNamesFromPreferences	Method will get the comma separated string of all the account id in the component preferences.	Method will get the comma separated string of all the bank names in the component preferences.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getBankNamesFromPreferences	Method will get the comma separated string of all the account id in the component preferences.	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getBankNamesFromPreferences	Method will get the comma separated string of all the account id in the component preferences.	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	getBankNamesFromPreferencesOnline	Method will get the comma separated string of all the bank names in the component preferences.	This method is used to fetch beneficiary account on basis of settlement ref no @param settlementReferenceNumber @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {<a href="#">@link SettlementExecutionService#executePreferenceOnline</a>} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt;will create a component preference and then execute the preference.</p> <p>&lt;br&gt; @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException</p>	<p>This method executes the component preference online.this will credit the settlement GL with necessary amount.If it throws the exception then corresponding executeDebitInstruction should catch it and mark the instruction as failed. @param componentPreferenceDto @param accountId @param debitAmount @param valueDate TODO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {<a href="#">@link SettlementExecutionService#executePreferenceOnline</a>} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt;will create a component preference and then execute the preference.</p> <p>&lt;br&gt; @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException</p>	<p>This method is invoked for Final Settlement Processing. NOTE: Account Closure Request is not being raised previously. @param sessionContext @param accountId @param closureDTO @param applicationContext @param finalSettlement @param accountTransaction @param ddaSettlementAdapter @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {<a href="#">@link SettlementExecutionService#executePreferenceOnline</a>} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt;will create a component preference and then execute the preference.</p> <p>&lt;br&gt; @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException</p>	<p>@see com.ofss.fc.app.dda.service.core.IDemandDepositTransactionService#issueMCToSelf(com.ofss.fc.app.context.SessionContext, java.lang.String, java.lang.String, com.ofss.fc.datatype.dto.MoneyDTO)</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {<a href="#">@link SettlementExecutionService#executePreferenceOnline</a>} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt;will create a component preference and then execute the preference.</p> <p>&lt;br&gt; @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException</p>	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service {<a href="#">@link SettlementExecutionService#executePreferenceOnline</a>} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt; will create a component preference and then execute the preference. &lt;br&gt; @param instruction @param amountForSettlement @param nsfTreatment @param @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the &lt;br&gt; settlement service {@link SettlementExecutionService#executePreferenceOnline} &lt;br&gt;This method is used in case the component preference does not exist. The settlement service in this case &lt;br&gt; will create a component preference and then execute the preference.</p> <p>&lt;br&gt; @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException</p>	<p>This method will be used for the execution of the chosen settlement mode. The method will in turn call the &lt;br&gt; settlement service {@link SettlementExecutionService#executePreferenceOnline} &lt;br&gt; @param accountId @param amountForSettlement @param eventId @param serviceCode @param instruction @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceOnline	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service <a href="#">{@link SettlementExecutionService#executePreferenceOnline}</a> This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference. @param context @param productCode @param eventCode @param instruction @param amountForSettlement @param treatmentType @param narration @param moduleReferenceNo @param messageType @param shortNarration @param transactionLiteral @param fromOfferId @param additionalParameters @return @throws FatalException	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service <a href="#">{@link SettlementExecutionService#executePreferenceOnline}</a> This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference. 
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchSettlementDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method would validate the new offer id should not be same as the restriction maintained for TD principal payout where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchSettlementDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method would validate the new offer id should not be same as the restriction maintained for TD maturity where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchSettlementDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method would validate the new offer id should not be same as the restriction maintained for Loan disbursement where the account number passed is beneficiary account number in Loan disbursement setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	fetchSettlementDetailsForDDAccount	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return	This method derives the Account Linkage details for an account which is being used by one or more term Deposit Accounts for TD Payout for its Principal or Interest component. It would provide the linked Term Deposit Accounts with respective Deposit Numbers. @param accountId @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceForMiscSettlement	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> } @param accountId @param amountForSettlement @param eventId @param serviceCode @param bsbCode TODO @return	Method used for miscellaneous credits. Miscellaneous credits are mostly for patch works. <p>The method is also used for salary credit through file upload. The employer account is debited in the file processing header and amount will be banked in an ledger. Then this method will be called to credit the employee account, debiting for the same ledger. </p>The method will call settlement to settle the credit amount. If the bsb belongs to an external branch, then settlement mode will be sent as payments. For internal accounts, the bsb must be sent as null.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	executePreferenceForMiscSettlement	This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service { <a href="#">@link SettlementExecutionService#executePreferenceOnline</a> } @param accountId @param amountForSettlement @param eventId @param serviceCode @param bsbCode TODO @return	Method used for miscellaneous debits. Miscellaneous debits are mostly for patch works. <p>The method is also used for salary credit through file upload. The employer account is debited in the file processing header and amount will be banked in an ledger. This method will be called to perform this action. </p>The method will call settlement to settle the credit amount. If the bsb belongs to an external branch, then settlement mode will be sent as payments. For internal accounts, the bsb must be sent as null.
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return	This method is used to create a sweep out instruction to a TD account via settlement. This method would create a component preference item where the provider account would be casa and the beneficiary account would be a TD account @param sessionContext @param sweepOutRequestDTO @return @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return	This method is used to create a sweep out instruction to an external account via settlement. This method would create a component preference item where the provider account would be casa and the beneficiary account would be an external account. @param sessionContext @param sweepoutRequestDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return	This method is used to create a sweep out instruction to a TD account via settlement. This method would create a component preference item where the provider account would be casa and the beneficiary account would be a TD account @param sessionContext @param sweepOutRequestDTO @return @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return	This method is used to create a sweep out instruction to an external account via settlement. This method would create a component preference item where the provider account would be casa and the beneficiary account would be an external account. @param sessionContext @param sweepoutRequestDTO @return @throws FatalException
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return	This method will fetch the component preference item in case of sweepout to a term deposit account. @param sweepoutToTDRequestDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDASettlementAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDASettlementAdapter	buildComponentPreferenceDTO	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return	This method will fetch the component preference item in case of sweepout to an external account. @param sweepOutRequestDTO @return
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchCommonServicesAdapter	fetchHostData		This method is used to fetch values like posting date, next posting date and previous posting date which is required by the UI channels to construct sessionContext @param com.ofss.fc.app.context.SessionContext,sessionContext,sessionContext @throws FatalException
IBranchCommonServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\branch\IBranchCommonServicesAdapter	fetchHostData		
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of outgoing clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method sends the request to raise the alerts for successful transaction of incoming clearing @param activityId @param accountNumber @param eventId @param cequeNumber @param valueDate @param chequeAmount @param dateOfTransaction
IChannelEventProcessingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelEventProcessingAdapter	persistActivityLog	This method registers activity and generates event for given activity id and event id @param applicationContext @param activityId @param eventId @param activityLog @throws FatalException	This method persists activity and event log for given activity and event id @param activityId @param eventId @param activityLog @param eventDate @throws FatalException
IContactPointAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IContactPointAdapter	updateContactPoint		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLoansAdapter	linkPremiumToExistingAccount		@param accountId @param valueDate @param premiumAmount @return @throws FatalException
IInsuranceLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLoansAdapter	linkPremiumToExistingAccount		@param insuranceDTO @param valueDate @throws FatalException
IInsuranceLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLoansAdapter	linkPremiumToExistingAccount		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
IInsuranceLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLoansAdapter	intimateLoanOnPolicyCancel		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
IInsuranceLoansAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\IInsuranceLoansAdapter	intimateLoanOnPolicyCancel		@param sessionContext @param insuranceInstanceLinkageDTO @return @throws com.ofss.fc.infra.exception.FatalException
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ITitleSearchAgentEligibilityAdapter	addRestrictionLinkageAndException		This method adds a new Restriction Linkage And Exception. @return transactionStatus @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ITitleSearchAgentEligibilityAdapter#addRestrictionLinkageAndException(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.commonservice.core.dto.restriction.RestrictionExceptionAdditionDTO)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ITitleSearchAgentEligibilityAdapter	fetchAllAllowableValues		Returns the <b>Eligible</b> titleSearchAgents This will call the Eligibility Framework with the collateral Facts, and the call will return the list of TitleSearchAgents that are applicable for the TitleSearch. @fcb.param SessionContext,sessionContext @fcb.param String,collateralID @return {@link com.ofss.fc.app.lcm.dto.collaterals.titlesearch.FetchAgentInquiryResponse} @throws FatalException
ITitleSearchAgentEligibilityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ITitleSearchAgentEligibilityAdapter	fetchAllAllowableValues		This method fetches all allowable values . @return AllAllowableValues @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ITitleSearchAgentEligibilityAdapter#fetchAllAllowableValues(com.ofss.fc.app.context.SessionContext, com.ofss.fc.enumeration.commonservice.DomainEntityType, com.ofss.fc.app.commonservice.core.dto.restriction.PreliminaryRestrictionDTO[])
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	validateAccountDetails	This method returns the rate types to be used for each settlement mode for a given event code and either product or module. If product is supplied, then the module is not taken into consideration. The order in which the modes are passed is used to returning back the rate types for each mode @param context @param productCode @param eventCode @param callerModule @param modeTypes @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	validateAccountDetails	This method returns the rate types to be used for each settlement mode for a given event code and either product or module. If product is supplied, then the module is not taken into consideration. The order in which the modes are passed is used to returning back the rate types for each mode @param context @param productCode @param eventCode @param callerModule @param modeTypes @return @throws FatalException	This method is used to perform account related checks.
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HIOSA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HISOA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HISOA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HISOA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	This method is called to validate the Hiso related validation.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HIOSA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	This method is used to validate wheather account transfer is allowed. @param fromAccount @param fromModuleType @param toAccount @param toModuleType
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HIOSA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction INDicates the direction of fund flow from Customer perspective @param transactionType INDicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isAccountTransferAllowedForHIOSA	This method will check if the HIOSA validations for funds transfer are being satisfied or not. AS per HIOSA, the money from High interest product account cannot move out. This is applicable to CASA products only for now. If the toAccount is null , it is treated as external account, else it will be our account @param fromAccount @param fromModuleType @param toAccount @param toModuleType @return	This method will process the settlement request for given execution context. It will call the appropriate module API based on the direction using the amount @param context INDicates the session context to be used for execution. This will also carry the predefined transaction reference number and the original reference number for reversal @param details The settlement mode details to be used @param direction INDicates the direction of fund flow from Customer perspective @param transactionType INDicates whether the current invocation should result in reversal. In such case, the context should carry the original reference number @param amount The amount to be settled
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isLoanAccountValid	This method is used to validate the loan account. @param loanAccId @return	
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	isLoanAccountValid	This method is used to validate the loan account. @param loanAccId @return	
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	The following method is used to fetch offer which is related to the provided account id. @param accountNumber @param accountModuleType @return
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	The following method is used to fetch offer which is related to the provided account id. @param accountNumber @param accountModuleType @return
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	The following method is used to fetch offer which is related to the provided account id. @param accountNumber @param accountModuleType @return
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	The following method is used to fetch the offer id using the account id. @param accountId @return @throws FatalException
ISettlementAccountValidationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementAccountValidationAdapter	fetchAccountRelatedOfferId	This method is used to fetch offer id which is related to the provided account. @param accountId @param accountType @return @throws FatalException TODO	The following method is used to fetch the offer id using the account id. @param accountId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IROSOProcessForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IROSOProcessForApprovalRequestAdapter	initiateROSOProcessForApprovalProcess		@see com.ofss.fc.app.collection.service.rosoprocessforapproval.IROSOProcessForApprovalApplicationService This method creates case from ROSOProcessForApprovalRequestDTO @param p_SessionContext @param p_ROSOProcessForApprovalRequestDTO @throws FatalException
IROSOProcessForApprovalRequestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IROSOProcessForApprovalRequestAdapter	initiateROSOProcessForApprovalProcess		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	isAdapterMocked	This methods checks whether adapter mocking is enabled ; If true, it returns true to adapter mocked. If false, it returns the adapter mocking flag for the adapter from the adapter factory properties. @param adapterMockConstant @return boolean	This method asserts required authorizations for overridable policies. @param sessionContext @param status
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		This is the primary service providing functionality for carrying out the authorizations that are configured as required against the service. The application service relies on the underlying domain services to provide the authorization functionality. @fcb.param MI,SessionContext,sessionContext, , Indicates the session Context. @fcb.param MI,String, roleId, Indicates the role Id. @return AuthorizationAssertionResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		This method asserts required authorizations for overridable policies. @param sessionContext @param status
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		This method delegates the calls to the appropriate methods during security access check process. First of all, if the service is blacked out for the period in which the user is accessing it, then he  will not be allowed to perform the transaction.    Access policies are maintained on OID server corresponding to roles.  Based on the policies, the user  has access on a particular service or not is validated.    If the service under execution is eligible for OAAM validations, call is made to OAAM.  OAAM validates the policies and rules set for the service and returns the action to be taken.  Action can be Allow, Block, Challenge1FA, Challenge1FADelay, Challenge2FA, Challenge2FADelay.    Next assertRequiredAuthorizations is used to test for business conditions that can be either rejected,  ignored or overridden. If status is marked as overridden, system fetches the configured workflow and   distributes workitems accordingly.    @param serviceId @param context @param serviceInputs @param parameterNames @throws FatalException
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		This method is used to test for business conditions that can be either rejected, ignored or overridden. If status is marked as overridden, system fetches the configured workflow and distributes workitems accordingly. @param sessionContext - SessionContext object @throws FatalException -

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Exception being returned
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizations		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	storeDefinedWorkflowsForSystemOverridableExceptions		This method sets the defined workflow in FCRThreadAttribute. @fcb.param MI,int,branchCode, , Indicates the branch code of the transaction branch. @throws FatalException
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	storeDefinedWorkflowsForSystemOverridableExceptions		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizationsForOverridablePolicies		This method asserts required authorizations for overridable policies. @param sessionContext @param status
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	assertRequiredAuthorizationsForOverridablePolicies		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	prepareAuthorizationServicesSingleton		Initialize all security related singletons @param sessionContext @throws FatalException
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	prepareAuthorizationConfigurationSingleton		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	prepareAuthorizationConfigurationSingleton		Initialize all security related singletons @param sessionContext @throws FatalException
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	prepareServiceRulesSingleton		
IAuthorizationAssertionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IAuthorizationAssertionAdapter	prepareServiceRulesSingleton		Initialize all security related singletons @param sessionContext @throws FatalException
IUisvcSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcSecurityAdapter	fetchSecurityServicesResponse		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IUisvcSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcSecurityAdapter	fetchSecurityServicesResponse		
IUisvcSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\uisvc\IUisvcSecurityAdapter	fetchSecurityServicesResponse		
IDDAArrearProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAArrearProductAdapter	fetchProductInformation	This method is used to call the pmu module to obtain product related information.This method is call to get longest overdue first and appropriate sequence.	
IDDAArrearProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAArrearProductAdapter	fetchProductInformation	This method is used to call the pmu module to obtain product related information.This method is call to get longest overdue first and appropriate sequence.	
IDDAArrearProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAArrearProductAdapter	fetchProductInformation	This method is used to call the pmu module to obtain product related information.This method is call to get longest overdue first and appropriate sequence.	
ISecurityDDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\ISecurityDDAAdapter	fetchBalanceDetails		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchSavingsAccountsInformation	@param applicationContext @param partyId	
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		This represents the fetchAllDemandDepositAccountsOfParty method of the entity.It fetches all demand deposit accounts where party is related with one of the following relationship.  1. Sole Owner   2. Joint and First   3. Joint and Others   4. Others - Affiliate.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,PartyToAccountRelationshipDTO,partyToAccountRelationshipDTO, ,partyToAccountRelationshipDTO @return {@link com.ofss.fc.app.party.dto.relation.account.PartyToAccountRelationshipInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Method to update the status of all the accounts held by the customer to Regular, in case customer is removed as Blacklisted. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Method to update the status of all the accounts held by the customer to Total Block, in case customer is Blacklisted. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Service to fetch all regular CASA accounts, TD accounts and external accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Service to fetch all CASA accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAccounts		Service to fetch all CASA accounts and TD accounts for a given party id. @param sessionContext @param partyId @return PartyAllAccountsInformationResponse @throws FatalException p

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Updates the account status to regular.Used when the KYCStatus changes to confirmed. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Updates the account status to Debit Block.Used when the newly added party is KYC unconfirmed. @param accountId of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Updates the account status to regular.Used when customer id deceased. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Updates the account status to regular.Used when customer id deceased. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Updates the account status to regular.Used when KYC is Unconfirmed. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Remove account status Debit Block.Used when the KYC unconfirmed party is deleted. @param accountId of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Method to update the status of all the accounts held by the customer to Regular, in case customer is removed as Blacklisted. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatus		Method to update the status of all the accounts held by the customer to Total Block, in case customer is Blacklisted. @param partyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchDDAInformation		
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Updates the account status to Debit Block.Used when the newly added party is KYC unconfirmed. @param accountId of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Updates the account status to regular.Used when KYC is Unconfirmed. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusRemoveDebitBlock		Remove account status Debit Block.Used when the KYC unconfirmed party is deleted. @param accountId of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusRemoveDebitBlock		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToBlock		Updates the account status to regular.Used when customer id deceased. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToBlock		Method to update the status of all the accounts held by the customer to Total Block, in case customer is Blacklisted. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Updates the account status to Debit Block.Used when the newly added party is KYC unconfirmed. @param accountId of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Updates the account status to regular.Used when KYC is Unconfirmed. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Call to adapter to update the Account Status to Debit Block if party added is KYC unconfirmed. @param partyAccountRelationship @throws FCRException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToDebitBlock		Method to update the status of all the accounts held by the customer to Regular, in case of KYCStatus Change. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToRegular		Updates the account status to regular.Used when customer id deceased. @param {@List} of DemandDepositAccounts. @see com.ofss.fc.domain.party.service.kyc.KYCHistoryService#updateKYCStatus(com.ofss.fc.domain.party.entity.core.IKYCHistory)
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	updateDDAStatusToRegular		Method to update the status of all the accounts held by the customer to Regular, in case customer is removed as Blacklisted. @param partyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	primaryAccountHolderIDMaintenance		Maintain primary Account Holder ID for CASA Account. @param accountId,newpartyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the Loan module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the TD module for the change in ownership.  @param accountId @param newPrimaryPartyId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	primaryAccountHolderIDMaintenance		If there is a change in primary ownership of account then it will update the DDA module for the change in ownership.  @param accountId @param newPrimaryPartyId
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchOperatingInstruction		
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchBalanceDetails	Method to fetch CASA and OD balances for a given account @param accountId @throws FatalException	Method to fetch CASA and OD balances for a given account @param accountId @throws FatalException
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchAverageBalanceHistory		Method to fetch average CASA balances for a given account @param accountId @throws FatalException
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	fetchLastChequeBookRequest		
ICustomerCASAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCASAAdapter	createChequeBookRequest		



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOperatingInstructionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOperatingInstructionAdapter	fetchAuthorizedPartySignatories		
IOperatingInstructionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IOperatingInstructionAdapter	fetchAuthorizedPartySignatories		
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event</p> <p>&lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B</p> <p>&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	<p>This method raises accounting in case of Cheque Deposit to CASA during a Multi Fund Deposit transaction. @fcb.param MI, TransactionContext, transactionContext &lt;li&gt;{@link com.ofss.fc.app.context.TransactionContext}&lt;/li&gt; @fcb.param MI, TransactionLogDTO, transactionLog &lt;li&gt;{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO }&lt;/li&gt; @fcb.param MI, AccountingEventAmountTagDetailsDTO, accountingEventAmountTagDetailsDTOArray &lt;li&gt;{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}&lt;/li&gt; @fcb.param MI, MessageLogItemDTO, messageLogItem &lt;li&gt;{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}&lt;/li&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event &lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B&lt;/br&gt; This method calls &lt;br&gt; <a href="#">@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)</a> which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	<p>This method raise the accounting entries. This method can be called as many times as the caller want. If the caller calls with the same eventId, the sequence number for the same event is incremented to facilitate the accounting framework requirement. If this method is called for the first time for the eventId, then this method get the first sequence number for the event with the existing transaction reference number.</p> <p>@fcb.param MI, List, accountingEventAmountTagDetailsDTOList @fcb.param MI, String, transactionReferenceNumber @fcb.param MI, String, transactionDescription @fcb.param MI, String, primaryAccountId @fcb.param MI, String, eventId @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event &lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	<p>This method raises the accounting for cash deposit event in case of multi fund deposit transaction.</p> <p>@fcb.param MI, ApplicationContext, applicationContext &lt;li&gt; {@link com.ofss.fc.framework.context.ApplicationContext}&lt;/li&gt; @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventDetailDTO &lt;li&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}&lt;/li&gt; @fcb.param MI, CashTransactionDTO, cashTransactionDTO &lt;li&gt; {@link com.ofss.fc.app.dda.dto.transaction.CashTransactionDTO}&lt;/li&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event &lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.  @param applicationContext  @param accountingEventDTO</p>	<p>This method will raise an accounting for the bank draft liquidation only. @fcb.param MI, TransactionContext, transactionContext &lt;li&gt;{@link com.ofss.fc.app.context.TransactionContext}&lt;/li&gt; @fcb.param MI, BankDraftDTO, bankDraftDTO &lt;li&gt;{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}&lt;/li&gt; @fcb.param MI, String, eventId @fcb.param MI, List, accountingEventAmountTagDetailsDTO &lt;li&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}&lt;/li&gt; @return accountTagDTOArray &lt;li&gt;{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}&lt;/li&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event &lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B&lt;/br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus. @param applicationContext @param accountingEventDTO</p>	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event &lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B&lt;/br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseAndProcessAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus. @param applicationContext @param accountingEventDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event</p> <p>&lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B</p> <p>&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService #raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event</p> <p>&lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B</p> <p>&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	<p>This method raises an accounting event for successful settlement payout transaction.&lt;br /&gt; It does the following,&lt;br /&gt; &lt;br /&gt; &lt;ol&gt; &lt;li&gt;Prepares an {@link AccountingEventDTO} via {@link SettlementInstructionPayoutApplication#prepareAccountingEventDTO} method.&lt;/li&gt; &lt;li&gt;Raise an accounting event via {@link IAccountAccountingAdapter#raiseAccountingEvent} method&lt;/li&gt; &lt;/ol&gt; @param applicationContext @param accountingEventAmountTagDetailsDTO @param ledgerCode @param branchCode @param accountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	<p>This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event</p> <p>&lt;br&gt; into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B</p> <p>&lt;br&gt; This method calls &lt;br&gt; {@link AccountingEventApplicationService#raiseNewAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus.</p> <p>@param applicationContext @param accountingEventDTO</p>	<p>&lt;p&gt;This method is to raise accounting for all the fees applicable on the event.As per the bank level parameter (show discount) will include the discount accounting . &lt;/p&gt; &lt;p&gt; Following are the required steps:&lt;br&gt; 1.populate accounting event DTO {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt; &lt;p&gt; 2. Get instance of Account accounting adapter &lt;br&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt; &lt;p&gt; 3. Call raise Accounting of adapter.&lt;br&gt; {@link com.ofss.fc.app.adapter.impl.account.AccountAccountingAdapter#raiseAccountingEvent}&lt;/p&gt; @param primaryAccountId @param accountingEventDetailDTO @param accrualStatusType @param taskModuleType @param eventCode @param transactionDescription @param transactionRefNo @param accountBranch @param productCode @param partyId @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseReverseAccountingEvent	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. This method prepares {@link AccountingBehaviourDTO} and following are set.&lt;br/&gt;accountingBehaviourDTO.setIsAuthorized(true);&lt;br/&gt;accountingBehaviourDTO.setIsAuthorized(true);&lt;br/&gt;accountingBehaviourDTO.setIsBalancingRequired(true);&lt;br/&gt; Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus. @param applicationContext @param sequenceNumber Sequence number of the current accounting entry @param transactionReferenceNumber Current transaction reference number @param originalTransactionReferenceNumber Original transaction reference number @param loanEventConstant Loan Event Constant</p>	<p>This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. This method prepares {@link AccountingBehaviourDTO} and following are set.&lt;br/&gt;accountingBehaviourDTO.setIsAuthorized(true);&lt;br/&gt;accountingBehaviourDTO.setIsBalancingRequired(true);&lt;br/&gt; Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseAndProcessReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus. @param applicationContext @param sequenceNumber Sequence number of the current accounting entry @param transactionReferenceNumber Current transaction reference number @param originalTransactionReferenceNumber Original transaction reference number @param loanEventConstant Loan Event Constant</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAndProcessReversalAccountingEvent	This method is used to raise and immediately process Reversal Accounting Event. It has to be invoked only when Accounting Event (which is being raised already) is processed. @param applicationContext @param sequenceNumber @param transactionReferenceNumber @param originalTransactionReferenceNumber @param loanEventConstant	This method is used for raising a reversal event for the transaction and event sequence no provided. Using the originalReferenceNo and originalEventSequenceNo, the details of the event to be reversed is obtained and a new reversal event is created. This method prepares {@link AccountingBehaviourDTO} and following are set. accountingBehaviourDTO.setIsAuthorized(true); accountingBehaviourDTO.setIsBalancingRequired(true);  Then it calls {@link AccountingEventApplicationService#com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseAndProcessReversalAccountingEventForTransaction(SessionContext, String, int, String, String, AccountingBehaviourDTO)} which will return the TransactionStatus. @param applicationContext @param sequenceNumber Sequence number of the current accounting entry @param transactionReferenceNumber Current transaction reference number @param originalTransactionReferenceNumber Original transaction reference number @param loanEventConstant Loan Event Constant
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAndProcessReversalAccountingEvent	This method is used to raise and immediately process Reversal Accounting Event. It has to be invoked only when Accounting Event (which is being raised already) is processed. @param applicationContext @param sequenceNumber @param transactionReferenceNumber @param originalTransactionReferenceNumber	This method is used to raise and immediately process Reversal Accounting Event. It has to be invoked only when Accounting Event (which is being raised already) is processed. @param applicationContext @param sequenceNumber @param transactionReferenceNumber @param originalTransactionReferenceNumber @param loanEventConstant

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			er @param loanEventConstant	
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). @See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @fcb.param MI, AccountingEventDTO, accountingEventDTO <li> <a href="#">@link com.ofss.fc.app.accounting.dto.AccountingEventDTO</a> </li>	This method raise the accounting entries. This method can be called as many times as the caller want. If the caller calls with the same eventId, the sequence number for the same event is incremented to facilitate the accounting framework requirement. If this method is called for the first time for the eventId, then this method get the first sequence number for the event with the existing transaction reference number. @fcb.param MI, List, accountingEventAmountTagDetailsDTOList @fcb.param MI, String, transactionReferenceNumber @fcb.param MI, String, transactionDescription @fcb.param MI, String, primaryAccountId @fcb.param MI, String, eventId @return
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseFinancialEvent	The method raiseAccountingEvent() allows to raise an accounting event identified by accountTransaction.getTransactionAttributes.getEventId() and account.getProduct.getId(). @See <a href="#">@link DomainAccountingAdapter#raiseAccountingEvent</a> @fcb.param MI, AccountingEventDTO, accountingEventDTO <li> <a href="#">@link com.ofss.fc.app.accounting.dto.AccountingEventDTO</a> </li>	The method is used to raise financial event. <a href="#">@link AccountingEventApplicationService#raiseAndProcessAccountingEvent(SessionContext, AccountingEventDTO)</a> )} @param sessionContext @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This method raises accounting in case of Cheque Deposit to CASA during a Multi Fund Deposit transaction. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, TransactionLogDTO, transactionLog <li>{@link com.ofss.fc.app.pc.dto.log.TransactionLogDTO }</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO, accountingEventAmountTagDetailsDTOArray <li>{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, MessageLogItemDTO, messageLogItem <li>{@link com.ofss.fc.app.pc.dto.log.MessageLogItemDTO}</li> @throws FatalException
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This method raise the accounting entries. This method can be called as many times as the caller want. If the caller calls with the same eventId, the sequence number for the same event is incremented to facilitate the accounting framework requirement. IfThis method is called for the first time for the eventId, then this method get the first sequence number for the event with the existing transaction reference number. @fcb.param MI, List, accountingEventAmountTagDetailsDTOList @fcb.param MI, String, transactionReferenceNumber @fcb.param MI, String, transactionDescription @fcb.param MI, String, primaryAccountId @fcb.param MI, String, eventId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This method raises the accounting for cash deposit event in case of multi fund deposit transaction. @fcb.param MI, ApplicationContext, applicationContext <li> {@link com.ofss.fc.framework.context.ApplicationContext}</li> @fcb.param MI, AccountingEventAmountTagDetailsDTO[], accountingEventDetailDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @fcb.param MI, CashTransactionDTO, cashTransactionDTO <li> {@link com.ofss.fc.app.dda.dto.transaction.CashTransactionDTO}</li> @throws FatalException
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This method will raise an accounting for the bank draft liquidation only. @fcb.param MI, TransactionContext, transactionContext <li>{@link com.ofss.fc.app.context.TransactionContext}</li> @fcb.param MI, BankDraftDTO, bankDraftDTO <li>{@link com.ofss.fc.app.pc.bankdraft.dto.BankDraftDTO}</li> @fcb.param MI, String, eventId @fcb.param MI, List, accountingEventAmountTagDetailsDTO <li> {@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @return accountTagDTOArray <li>{@link com.ofss.fc.app.accounting.dto.AccountingEventAmountTagDetailsDTO}</li> @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This represents the raiseAccountingEvent method which raises the accounting event occurred. After getting the required details for processing of accounting entries, this method will persist the input values of accounting event   into event table - FLX_DA_ACCT_EVENT_B which will have one to one relationship with behaviour - FLX_DA_ACCT_BEHAVIOUR_B and one to many relationship with details - FLX_DA_ACCT_EVENT_DTLS_B</br> This method calls   {@link AccountingEventApplicationService#raiseAndProcessAccountingEvent(com.ofss.fc.app.context.SessionContext, AccountingEventDTO)} which will return the TransactionStatus. @param applicationContext @param accountingEventDTO
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	This method raises an accounting event for successful settlement payout transaction.  It does the following,    <ol> <li>Prepares an {@link AccountingEventDTO} via {@link SettlementInstructionPayoutApplication#prepareAccountingEventDTO} method.</li> <li>Raise an accounting event via {@link IAccountAccountingAdapter#raiseAccountingEvent} method</li> </ol> @param applicationContext @param accountingEventAmountTagDetailsDTO @param ledgerCode @param branchCode @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				accountId
IAccountAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountAccountingAdapter	raiseAccountingEvent	This method will raise accounting event after successful account transaction. @fcb.param MI, PCAccountingEventDTO, accountingRequest <li> {@link com.ofss.fc.app.adapter.pc.dto.PCAccountingEventDTO}</li> @throws FatalException	<p>&lt;p&gt;This method is to raise accounting for all the fees applicable on the event.As per the bank level parameter (show discount) will include the discount accounting . &lt;/p&gt; &lt;p&gt; Following are the required steps:&lt;br&gt; 1.populate accounting event DTO {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt; &lt;p&gt; 2. Get instance of Account accounting adapter &lt;br&gt; {@link com.ofss.fc.app.accounting.dto.AccountingEventDTO}&lt;/p&gt; &lt;p&gt; 3. Call raise Accounting of adapter.&lt;br&gt; {@link com.ofss.fc.app.adapter.impl.account.AccountAccountingAdapter#raiseAccountingEvent}&lt;/p&gt; @param primaryAccountId @param accountingEventDetailDTO @param accrualStatusType @param taskModuleType @param eventCode @param transactionDescription @param transactionRefNo @param accountBranch @param productCode @param partyId @throws FatalException</p>
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlsearch\ICollateralTitleSearchAdapter	processNewRequest		@param sessionContext @param collateralValuationDTO @param collateralValuation @throws FatalException @throws RuntimeException
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlsearch\ICollateralTitleSearchAdapter	processNewRequest		This message processes a new request @see com.ofss.fc.app.adapter.lcm.collaterals.titlsearch.ICollateralTitleSearchAdapter#processNewRequest(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.titlsearch.CollateralTitleSearchObject)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchAdapter	processAmendRequest		
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchAdapter	processAmendRequest		This method processes amend request. @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ICollateralTitleSearchAdapter#processAmendRequest(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.titlesearch.CollateralTitleSearchObject)
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchAdapter	processCancelRequest		This method <u>Cancels</u> the TitleSearch request for the given request message reference . @fcb.param SessionContext,sessionContext, Holds the basic information about the user of the system. @fcb.param CollateralTitleSearchCancellationDTO,cancellationDTO @return {@link com.ofss.fc.service.response.TransactionStatus} An instance   of TransactionStatus contains the information whether the transaction is success\failure.   @throws FatalException
ICollateralTitleSearchAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ICollateralTitleSearchAdapter	processCancelRequest		This method processes a cancellation request. @see com.ofss.fc.app.adapter.lcm.collaterals.titlesearch.ICollateralTitleSearchAdapter#processCancelRequest(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.lcm.dto.collaterals.titlesearch.CollateralTitleSearchCancellationDTO)



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToPartyRelationshipForAddOnCardHolder	This method is used to add party to party relationships for add on card holders @param sessionContext @param addOnCardHolderDTOs @return PartyToPartyRelationshipDTO[] @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToPartyRelationshipForAddOnCardHolder	This method is used to add party to party relationships for add on card holders @param sessionContext @param addOnCardHolderDTOs @return PartyToPartyRelationshipDTO[] @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToPartyRelationshipForAddOnCardHolder	This method is used to add party to party relationships for add on card holders @param sessionContext @param addOnCardHolderDTOs @return PartyToPartyRelationshipDTO[] @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToPartyRelationshipForAddOnCardHolder	This method is used to add party to party relationships for add on card holders @param sessionContext @param addOnCardHolderDTOs @return PartyToPartyRelationshipDTO[] @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForPrimaryCardHolder	This method is used to create or update party to account relationship for primary card holder @param sessionContext @param accountId @param partyId @return TransactionStatus @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology.This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference.After that it will save the current state of simulation.Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForPrimaryCardHolder	This method is used to create or update party to account relationship for primary card holder @param sessionContext @param accountId @param partyId @return TransactionStatus @throws FatalException	The method createOrUpdateCreditCardAccountDetails is exposed for external system to create or update credit card account in OBP in async flow  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForPrimaryCardHolder	This method is used to create or update party to account relationship for primary card holder @param sessionContext @param accountId @param partyId @return TransactionStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForAddOnCardHolder	This method is used to create or update party to account relationship for add on card holder @param sessionContext @param accountId @param addOnCardHolderDTO @param existingPartyToAccountRelationships @return TransactionStatus @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForAddOnCardHolder	This method is used to create or update party to account relationship for add on card holder @param sessionContext @param accountId @param addOnCardHolderDTO @param existingPartyToAccountRelationships @return TransactionStatus @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForAddOnCardHolder	This method is used to create or update party to account relationship for add on card holder @param sessionContext @param accountId @param addOnCardHolderDTO @param existingPartyToAccountRelationships @return TransactionStatus @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	isKYCConfirmed	This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @throws FatalException	This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	isKYCConfirmed	This method is responsible for calling the party module in order to get the KYC status of the party. It works as follows: <li>Call {@link KYCHistoryApplicationService#fetchKYCCheckStatusForRole}. party Id is as passed and the role is Customer by default</li> <li>This method returns KYCHistoryInquiryResponse.KYCHistoryDTO as output</li> <li>if kycHistoryDTO.KYCStatus == Confirmed && kycHistoryDTO.kycExpiryDate is greater than the posting date then return true </li>Else return false indication KYC is not done.	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@throws FatalException	
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	The method originateCreditCard is responsible for actually doing credit card request   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology. This service will also save the card reference number received from the external card system against the party id of the card holder in OBP system for future reference. After that it will save the current state of simulation. Then finally it does state transition for credit card application from its current state to CREDIT_CARD_CREATED.   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO,creditCardDetailsDTO,, This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	The method createOrUpdateCreditCardAccountDetails is exposed for external system to create or update credit card account in OBP in async flow @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardAccountInputDTO, creditCardAccountInputDTO, , This class holds application id, facility id and account configuration DTO @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link CreditCardAccountResponse} This class contains the information such as card account number and configuration DTO. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToAccountRelationship	Publish party to account relationship @param sessionContext @param accountId @return @throws FatalException	
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToPartyRelationship	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToPartyRelationship	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToPartyRelationship	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToPartyRelationship	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	publishPartyToPartyRelationship	Publish party to party relationship @param sessionContext @param partyRelationshipDTO @return @throws FatalException	
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchPartyFinancialProfileForParty	This service will fetch the party's current financial profile @throws FatalException	
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchAddonPartiesForAccount	This method is used to fetch add on parties for credit card account @param sessionContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException	
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchAddonPartiesForAccount	This method is used to fetch add on parties for credit card account @param sessionContext @param accountId @return PartyToAccountRelationshipDTO[] @throws FatalException	Populate account facility details to Credit card details DTO @param sessionContext @param creditCardDetailsDTO @param creditCardAccountDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchPartiesOfAccount	This method is used to fetch parties for account @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	The method linkAddOnCardHolder is responsible for actually linking add on card holder to an existing credit card   The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, ,This attribute holds all the personal information of an add-on card holder as well his card account preferences. @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchPartiesOfAccount	This method is used to fetch parties for account @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	The method updateLinkAddOnCardHolder is exposed to external credit card system to update the link add card holder details in OBP cards module in async flow The actual implementation of credit card will be based on runtime configuration and will be done via adapter methodology   @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,AddOnCardHolderDTO,addOnCardHolderDTO, This attribute holds all the personal information of an add-on card holder as well his card account preferences. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link com.ofss.fc.service.response.TransactionStatus} An instance of the transaction status which contains the required information about the status of execution indicating if the invocation succeeded or failed. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchPartiesOfAccount	This method is used to fetch parties for account @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	fetchPartiesOfAccount	This method is used to fetch parties for account @param sessionContext @param accountId @return AccountHolderInquiryResponse @throws FatalException	This method would validates Balance Transfer Data Inputs. <ol> It performs following validation : <li>Check if the Add On Card Holders applied are more than Maximum applicable for Offer.</li> <li>Internally it will call validate each instance of AddOnCardHolder using com.ofss.fc.app.card.service.credit.policy.AddonCardHolderBusinessPolicy.validateAddonCardHolderDetails</li> </ol> @param sessionContext @param accountConfigurationDTO @param addOnCardHolderDTO
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForPrimaryAndAddon	This method is used to create or update party to account relationship for primary and add on card holders @param sessionContext @param accountId @param primaryCardHolderId @param addOnCardHolderDTOs @return @throws FatalException	The method originateExternalCreditCard is responsible for originating external credit card in OBP (Cards, LCM and party module).  The actual implementation of credit card will be based on credit card details fetched from the external system.  @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,CreditCardDetailsDTO, creditCardDetailsDTO, , This class holds the credit card details from the external system @return {@link OriginateExternalCreditCardResponse} This class contains the information such as card account number and facility id created. @throws FatalException
ICardPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\card\ICardPartyAdapter	addPartyToAccountRelationshipForPrimaryAndAddon	This method is used to create or update party to account relationship for primary and add on card holders @param sessionContext @param accountId @param primaryCardHolderId @param addOnCardHolderDTOs @return @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.IInformationLog,informationLog, ,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeName	This method fetch the counterparty details on the bases of partyId of own customer and the counterparty account number @param partyId @param accountId @return	
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.IInformationLog,informationLog, ,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			@param institutionId @return	
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePaymentName	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquireBillerDetails	This method is to fetch biller details on the basis of biller ID. @param billerId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.IInformationLog,informationLog, ,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquireBillerCustomerRegistrationDetails	This method fetches the details of the biller customer registration, if the given biller is registered by the given party. @param billerId @param partyId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.IInformationLog,informationLog, ,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePaymentDetails	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.IInformationLog,informationLog, ,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquirePayeeDetails	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return
IChannelPaymentsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelPaymentsAdapter	inquireCounterPartyPayeeDetails	This method fetch the counter-party details on the bases of partyId of own customer, the counter-party account and Institution Id number @param partyId @param accountId @param institutionId @return	This represents the fromInformationLog method of the entity. @fcb.param MI,com.ofss.fc.domain.channel.entity.log.InformationLog,informationLog,informationLog TODO: Explain the informationLog above @return {@link InformationLogDTO} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
ICustomerCreditAccountsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\ICustomerCreditAccountsAdapter	fetchCreditAccountsInformation	@param applicationContext @param partyId	@return Object
IGEFUPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUPCAAdapter	execute	@param sessionContext @param fundExchangeRequestDTO @return @throws FatalException	
IGEFUPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUPCAAdapter	execute	@param sessionContext @param fundExchangeRequestDTO @return @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler
IGEFUPCAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\upload\IGEFUPCAAdapter	execute	@param sessionContext @param fundExchangeRequestDTO @return @throws FatalException	@see com.ofss.fc.upload.AbstractRecordHandler

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
	ter			
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	convertMoney	This method will make a call to the common currency service to convert the variance into price currency if the currencies are different. @return @throws FatalException	This method will make a call to the common currency service to convert the variance into price currency if the currencies are different. @return @throws FatalException
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	convertMoney	This method will make a call to the common currency service to convert the variance into price currency if the currencies are different. @return @throws FatalException	This method returns the following on the basis of conditions mentioned; This method will first check if both the money objects are within comparable money are of same currency; or else will convert them to local currency and them compare. -1 -- if the monetaryValue of this amount is less than the monetaryValue of the amount passed. 0 -- if the monetaryValue of this amount is equal to the monetaryValue of the amount passed. 1 -- if the monetaryValue of this amount is greater than the monetaryValue of the amount passed. @param ComparableMoney comparableMoney @return int
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	convertMoney	This method will make a call to the common currency service to convert the variance into price currency if the currencies are different. @return @throws FatalException	This method will make a call to the common currency service to convert the variance into price currency if the currencies are different. @param money @param currency @return Money
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	fetchLocalCurrencyCode	This method will return local currency of the bank. @param bankCode @return @throws FatalException	This method will checks if the money object passed to it is in a different currency than that of the local currency of the bank. Will return a boolean @param money @return boolean

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	fetchLocalCurrencyCode	This method will return local currency of the bank. @param bankCode @return @throws FatalException	This method returns the following on the basis of conditions mentioned; This method will first check if both the money objects are within comparable money are of same currency; or else will convert them to local currency and them compare. -1 -- if the monetaryValue of this amount is less than the monetaryValue of the amount passed. 0 -- if the monetaryValue of this amount is equal to the monetaryValue of the amount passed. 1 -- if the monetaryValue of this amount is greater than the monetaryValue of the amount passed. @param ComparableMoney comparableMoney @return int
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	processAndClearPendingRuleHistoryItems	The method will process any rule execution history items in the session and clear the same from thread local.	
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	processAndClearPendingRuleHistoryItems	The method will process any rule execution history items in the session and clear the same from thread local.	
IRuleAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\rule\IRuleAdapter	processAndClearPendingRuleHistoryItems	The method will process any rule execution history items in the session and clear the same from thread local.	
InsuranceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ins\collection\InsuranceCollectionAdapter	updateInsurance	@param feederInstrDtlsWrapperDTO @throws FatalException	This method is responsible to update the FeederInsrDtlsWrapperDTO with the insurance details @param feederInstrDtlsWrapperDTO instance of FeederInsrDtlsWrapperDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IInsuranceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\collection\IInsuranceCollectionAdapter	updateInsurance	@param feederInstrDtlsWrapperDTO @throws FatalException	This method updates the collection module on the changes in the insurance details @param insuranceDTO @param insuranceLinkageDTO @param recordType
IInsuranceCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ins\collection\IInsuranceCollectionAdapter	updateInsurance	@param feederInstrDtlsWrapperDTO @throws FatalException	
IChannelSMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\channel\IChannelSMSAdapter	fetchUser	This method is used for fetching the business unit for given user id @param userId	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	This method will return the primary account holder of the account, @param sessionContext @param accountId @param accountType @return String

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountForLinkage	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException	This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountForLinkage	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException	@param sessionContext @param offsetAccountDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountForLinkage	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException	This represents the updateOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountForLinkage	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException	This represents the reopenOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountForLinkage	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException	This method calls validateAccountForLinkage() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param offsetAccountCurrency @param linkageStartDate @param linkageExpiryDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	populateDefaultLinkageDates	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException	This represents the reopenOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	populateDefaultLinkageDates	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param accountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	populateDefaultLinkageDates	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException	This method calls populateDefaultLinkageDates() of LoanPrincipalOffsetApplication @param applicationContext @param loanAccountId @param productId @param offerId @return OffsetLinkageDatesResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createLoanDisbursementInstruction	This method calls the addMultipleLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService for setting up the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This methods setup process for settlement Instructions for a Application ,Facility or a Account level. Account level facility can be for loan account or OD. Facility level instruction again can be for an OD or loan Account. Application level can be done from origination.This method first validates the input coming from the client.All mandatory inputs need to send from the cliennt.It will then calls the method generateInstructionId to generates a unique instruction Id.This instruction id will be same for an item. setComponentPreferenceId method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				preference later. Business policy will be validated. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createLoanDisbursementInstruction	This method calls the addMultipleLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService for setting up the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This method will update the settlement payout instruction already maintained for an account or for a facility. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createLoanContributionPayinInstruction	This method calls the addLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService for setting up the Loan Contribution instruction. @param contributionInstructionDTO @throws FatalException	This methods set up the loan contribution Instruction. @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createLoanContributionPayinInstruction	This method calls the addLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService for setting up the Loan Contribution instruction. @param contributionInstructionDTO @throws FatalException	This methods setup process for settlement Instructions for a Application ,Facility or a Account level. Account level facility can be for loan account or OD. Facility level instruction again can be for an OD or loan Account. Application level can be done from origination.This method first validates the input coming from the client.All mandatory inputs need to send from the cliennt.It will then calls the method generateInstructionId to generates a unique instruction Id.This instruction id will be same for an item. setComponentPreferenceId method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. Business policy will be validated. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createLoanContributionPayinInstruction	This method calls the addLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService for setting up the Loan Contribution instruction. @param contributionInstructionDTO @throws FatalException	This method will update the settlement payout instruction already maintained for an account or for a facility. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method will process the disbursement instructions for the settlement payout instruction.  It will interact with Loan module for the processing of the disbursement instruction via  AccountLoanAdapter.  @param applicationContext @param accountId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This methods does auto disbursement by reading details from disbursement schedule that is set for the given account. @param applicationContext @param accountId @param disbursementDate
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This methods does auto disbursement As per data provided in LoanAutoDisbursementData. @param applicationContext @param accountId @param disbursementData @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method does the Subsequent Disbursement for the loan Account. @param sessionContext @param accountId @param scheduleTypeCode @param definitionDate @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method implements the process method for Auto Disbursement Action on the account. @param context @param accountId @param actionDate @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method do the Auto Disbursement For SplitLoan @param applicationContext @param accountId @param splitLoanAccountInstruction @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doAutoDisbursement	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doContributionProcess	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method will process the disbursement instructions for the settlement payout instruction.  It will interact with Loan module for the processing of the disbursement instruction via  AccountLoanAdapter.  @param applicationContext @param instructionId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doContributionProcess	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method will process the contribution instructions for the settlement payout instruction.  It will interact with CASA module for the processing of the contribution instruction via  AccountDDAAdapter.  @param applicationContext @param instructionId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doContributionProcess	This method will call loans for auto disbursement of the disbursement instructions  w.r.t the settlement payout instruction.  @param context @param accountId @param actionDate @throws FatalException	This method will call loans for process of contribution instructions  w.r.t the settlement payout instruction.  @param context @param instructionId @param actionDate @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will process the settlement instruction in the batch(EOD/BOD) and  call LOAN/CASA for settlement payout instruction, depending upon the instruction.  Step 1 : get the instruction id from the batch data.  Step 2 : check whether the execution of the contribution payin instruction w.r.t  settlement payout instruction is complete or not. In case the execution status of  the contribution payin instruction is not complete, then the execution of settlement  payout instruction will be marked as FAILED and the execution of the instruction will  stop there only. In case the execution status of the contribution payin instruction is  complete, go to step 3.  Step 3 : process the disbursement instruction, based on the account type of the  disbursement instruction.  Step 4 : process the settlement payout instruction.  @param applicationContext @param batchData @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will check whether execution of the contribution instruction for the  settlement payout instruction is complete or not.  In case the execution is not complete, the method will mark the execution status of  settlement instruction as FAILED, else it would start processing of the disbursement  instructions.  @param instructionId @return boolean @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	<p>This method will check if the bundle subscription fee has been collected and update the bundle benefit flag accordingly.  It will be used for link account cases. </p> @param bundleReferenceId @param isBundleBenefitApplicable @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This methods update the Contribution Received towards loan. @param accountId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will update the settlement payout instruction already maintained for an account or for a facility. @param sessionContext @param settlementPayoutInstructionDTO @return settlementPayoutInquiryResponse @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method delete the settlement payout instruction maintained for a account, facility or application level . @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method prepares the LoanContributionPayinInstructionDTO from contributionInstruction for setting up the loan contribution instruction. @param contributionInstructionItem @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method prepares the LoanDisbursementInstructionDTO from DisbursementInstructionItem for setting up the loan Disbursement instruction. @param disbursementInstructionItem @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method prepares the DDAContributionPayinInstructionDTO from contributionInstruction for setting up the CASA contribution instruction. @param contributionInstructionItem @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method prepares the LoanDisbursementInstructionDTO from DisbursementInstructionItem for setting up the loan Disbursement instruction. @param disbursementInstructionItem @param outlayDTOs @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will log the settlement instruction action in actions table.  For every settlement payout instruction item, an action will be logged in actions  table with the instruction id and execution date(scheduled and actual). @param settlementPayoutInstruction @param loanAdapter
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method will delete the settlement instruction action in actions table.  In case of update this will delete the previous uninitiated instruction from the action table. @param settlementPayoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	This method does the reversal of settlement payout transaction. @param sessionContext @param settlementPayoutSetupId @return transactionStatus @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getExecutionStatus	This method returns the execution status of the contribution payin instruction id  from Loan for settlement payout instruction processing.  @param accountId @param actionDate @throws FatalException	@param sessionContext @param instructionId @param actionDate @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanDisbursementInstruction	This method calls the updateLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService of Loan module for updating the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This method calls the update the LoanDisbursementInstruction method of class LoanDisbursementInstructionApplicationService for setting up the Loan Disbursement Instruction . @param loanDisbursementInstructionArray
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanDisbursementInstruction	This method calls the updateLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService of Loan module for updating the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param applicationContext @param settlementPayoutInstruction

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	deleteLoanDisbursementInstruction	This method calls the updateLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService of Loan module for updating the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This method calls the delete the LoanDisbursementInstruction method of class LoanDisbursementInstructionApplicationService for setting up the Loan Disbursement Instruction instruction. @param loanDisbursementInstructionArray
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	deleteLoanDisbursementInstruction	This method calls the updateLoanDisbursementInstruction method of LoanDisbursementInstructionApplicationService of Loan module for updating the Loan Disbursement instruction. @param loanDisbursementInstructionArray @throws FatalException	This method delete the settlement payout instruction maintained for a account, facility or application level. @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanContributionPayinInstruction	This method calls the updateLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService of loans module for updating the Loan Contribution instruction. @param loanContributionPayinInstructionDTO @throws FatalException	This methods update the loan contribution Instruction. @param accountId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateContributionReceived	This methods update the Contribution Received towards loan. @param accountId @throws FatalException	This methods update the Contribution Received towards loan. @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateContributionReceived	This methods update the Contribution Received towards loan. @param accountId @throws FatalException	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateContributionReceived	This methods update the Contribution Received towards loan. @param accountId @throws FatalException	This method executes the settlement payout instruction and raises an accounting event for the same.   It does the following,    <ol> <li>It loads {@link SettlementPayoutItem} via {@link SettlementPayoutInstructionItemsRepository} to obtain settlement reference number (i.e componentPreferenceId)</li> <li>Executes Settlement Instruction via {@link IAccountSettlementAdapter#executePreference} method.</li> <li>If execution of settlement instruction is successful (settlementExecutionResponse.getSummary().getUnsettledAmount().isZero()), it raises an accounting event via {@link SettlementInstructionPayoutApplication#raiseAccounting} method else thrown a {@link BusinessException}</li> </ol> @param applicationContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	deleteLoanContributionPayinInstruction	This method calls the updateLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService of loans module for updating the Loan Contribution instruction. @param loanContributionPayinInstructionDTO @throws FatalException	This methods delete the loan contribution Instruction. @param accountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	deleteLoanContributionPayinInstruction	This method calls the updateLoanContributionPayinInstruction method of loanContributionPayinInstructionApplicationService of loans module for updating the Loan Contribution instruction. @param loanContributionPayinInstructionDTO @throws FatalException	This method delete the settlement payout instruction maintained for a account, facility or application level . @param sessionContext @param settlementPayoutInstructionDTO @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @param linkageStartDate @param linkageEndDate @throws FatalException	This represents the deleteOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @param linkageStartDate @param linkageEndDate @throws FatalException	This method is used to log an offset maintenance action for Loan account @param offset



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @param linkageStartDate @param linkageEndDate @throws FatalException	This represents the closeOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @param linkageStartDate @param linkageEndDate @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @param linkageStartDate @param linkageEndDate @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @throws FatalException	This represents the deleteOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @throws FatalException	This method is used to log an offset maintenance action for Loan account @param offset
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @throws FatalException	This represents the closeOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestComputationAction	This method is used to log an interest computation action on offset maintenance @param beneficiaryAccountId @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	createAndConfigureLoanAccount	@param loanAccountConfigurationDTO @return	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchInstallmentRule		This method validates that parent loan account cannot have UDI stage. Consolidating other loan account into a loan account will result in increase in the outstanding balance of the parent account leading to a mismatch between the user defined schedule and the actual outstanding balance in the parent account. @param parentAccountId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	collectLegalorOutFeeforLoan		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processLoanOfferSwap	@param applicationContext @param eventId @param loanOfferSwapDetailsDTO @return @throws FatalException	<p>/&lt;p&gt;This method accepts the input dto {@link com.ofss.fc.app.loan.dto.offerswap.LoanOfferSwapDetailsDTO} to perform the offer swap action for Loan accounts. It calls for checking the mandatory inputs and assembles the dto to the domain object and delegates call to the domain service. &lt;/p&gt; Mandatory inputs for LOAN offer swap are: &lt;br&gt; &lt;ol&gt; &lt;li&gt;Account number cannot be null&lt;br&gt; &lt;li&gt;New offer id cannot be null FC_LN_1847&lt;br&gt; &lt;li&gt;Old offer id cannot be null FC_LN_1846&lt;br&gt; &lt;li&gt;Offer swap reason cannot be null FC_LN_1601&lt;/li&gt; &lt;/ol&gt; @param sessionContext @param eventId @param loanOfferSwapDetailsDTO @return @throws</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	changeOverDraftAccountToTermLoan	@param applicationContext @param eventId @param changeOverDraftToTermLoanRequestDTO @return @throws FatalException	This method is not usable. @param sessionContext @param changeOverDraftToTermLoanRequestDTO @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	changeTermLoanAccountToOverdraft	@param applicationContext @param eventId @param changeTermLoanToOverDraftRequestDTO @return @throws FatalException	This method is not usable. @param sessionContext @param changeOverDraftToTermLoanRequestDTO @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchAccountDetailsForPayoutInstruction	@param applicationContext @param accountId @return @throws FatalException	@param applicationContext @param accountId @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchAccountDetailsForPayoutInstruction	@param applicationContext @param accountId @return @throws FatalException	This method will fetch all the details related to facility Id.This will be called when user want to maintain a facility-level settlement payout instruction. call will be given to LCM module to inquire all the details for the selected facility ID. @param sessionContext @param facilityId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchActiveArrearSummaryByType	This method fetches the active arrears for the account summarized by arrear type. It calls LoanArrearManagementApplication.fetchActiveArrearSummaryByType(). @param applicationContext @param accountId @return	Provides the list of all active arrears which is not paid or partially paid.List of account id's is supplied by the caller.Total Due amount of each arrears type related to the account is provided back to the caller. @param accountList

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchActiveArrearSummaryByType	This method fetches the active arrears for the account summarized by arrear type. It calls LoanArrearManagementApplication.fetchActiveArrearSummaryByType(). @param applicationContext @param accountId @return	This method fetches the active arrears for the account summarized by arrear type. It calls LoanArrearManagementApplication.fetchActiveArrearSummaryByType(). @param applicationContext @param accountId @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	writeOffArrearPartially	This method calls LoanArrearManagementApplication.writeOffArrearPartially(). @param applicationContext @param eventId @param accountId @param arrearsByTypeDTOs @param reason1 @param reason2 @param provisionDrawdownAmount @return	This method is used to only in case of writing off of an arrear. Partial writing off of an arrear is required when the amount available for appropriation is less than the due amount. In this case the Flag which identifies whether the arrear is paid off or not is still N this in a way means that the due amount is not Zero . The due amount is being updated in case of partial appropriation of an arrear. <ArrearDTO> arrearListToBeUpdated <p> <span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt"></span><span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt">Data:</span> </p> <p> <span style="FONT-SIZE: 10pt">Provide The following Values:</span> </p> <p> <span style="FONT-SIZE: 10pt"> <table border="1"> <tbody> <tr> <td>Class : Ways to write of an arrear</td> <td>Arrear current write off Amount</td> <td>Due Amount</td> <td>Updated Due Amount</td> <td>Is Arrear Paid Off</td> </tr> <tr> <td>Case:1</td> <td>300AUD</td> <td>500AUD</td> <td>200AUD</td> <td>NO</td> </tr> <tr> </tr> @param applicationContext @param eventId @param accountId @param arrearWriteoffRequest @param reason1 @param reason2 @param provisionDrawdownAmount @param recoveryIneligibleAmount TODO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	writeOffArrearPartially	This method calls LoanArrearManagementApplication.writeOffArrearPartially(). @param applicationContext @param eventId @param accountId @param arrearsByTypeDTOs @param reason1 @param reason2 @param provisionDrawdownAmount @return	This method calls LoanArrearManagementApplication.writeOffArrearPartially(). @param applicationContext @param eventId @param accountId @param arrearsByTypeDTOs @param reason1 @param reason2 @param provisionDrawdownAmount @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	writeOffArrearFully	This method calls LoanArrearManagementApplication.writeOffArrearFully(). @param applicationContext @param eventId @param reason1 @param reason2 @param provisionDrawdownAmount @return	This method is used to only in case of writing off of an arrear. Fully writing off of an arrear is required when the amount available for appropriation is greater than the due amount. In this case the Flag which identifies whether the arrear is paid off or not is set as Y this in a way means that the due amount is Zero . The due amount is being updated in case of partial appropriation of an arrear. <ArrearDTO> arrearListToBeUpdated <p> <span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt"></span><span style="FONT-FAMILY: Book Antiqua; FONT-SIZE: 12pt">Data:</span> </p> <p> <span style="FONT-SIZE: 10pt">Provide The following Values:</span> </p> <p> <span style="FONT-SIZE: 10pt"> <table border="1"> <tbody> <tr> <td>Class : Ways to write of an arrear</td> <td>Arrear current write off Amount</td> <td>Due Amount</td> <td>Updated Due Amount</td> <td>Is Arrear Paid Off</td> </tr> <tr> <td>Case:1</td> <td>500AUD</td> <td>200AUD</td> <td>0</td> <td>YES</td> </tr> <tr> </tr> @param applicationContext @param eventId @param accountId @param reason1 @param reason2 @param currentPrincipalBal @param provisionDrawdownAmount @param recoveryIneligibleAmount TODO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	writeOffArrearFully	This method calls LoanArrearManagementApplication.writeOffArrearFully(). @param applicationContext @param eventId @param reason1 @param reason2 @param provisionDrawdownAmount @return	This method calls LoanArrearManagementApplication.writeOffArrearFully(). @param applicationContext @param eventId @param reason1 @param reason2 @param provisionDrawdownAmount @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	doReschedule	This method calls LoanRepaymentScheduleApplicationService.doReschedule() LoanRepaymentScheduleApplicationService.generateRepaymentRecordsOnReschedule() LoanRepaymentScheduleApplicationService.saveScheduleOnReschedule() @param sessionContext @param accountId @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param reasonCode @param reasonDesc @return	This method calls LoanRepaymentScheduleApplicationService.doReschedule() LoanRepaymentScheduleApplicationService.generateRepaymentRecordsOnReschedule() LoanRepaymentScheduleApplicationService.saveScheduleOnReschedule() @param sessionContext @param accountId @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param reasonCode @param reasonDesc @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	<p>This method will inquire the prices to be charged on bundle maintenance fee charging date when there is no nominated account to collect the fee from. It will inquire all the fees and call fee management service to update the fee as derffered. </p> @param applicationContext @param partyBundleLinkage
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	<p> Method does maintenance fee processing for account modules </p> <p> It fetches the account basic details having information of party, currency etc. </p> <p> For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data </p> @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	Method to process Maintenance fees from External account @param applicationContext @param accountBasicDetailsDTO @param bundleFeeBatchData @param componentPreferenceDTO @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	<p> Method does maintenance fee processing for account modules </p> <p> It fetches the account basic details having information of party, currency etc. </p> <p> For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data </p> @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	Method to process Maintenance fees from External account @param applicationContext @param accountBasicDetailsDTO @param bundleFeeBatchData @param componentPreferenceDTO @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	<p>This method will inquire the prices to be charged on bundle maintenance fee charging date when there is no nominated account to collect the fee from. It will inquire all the fees and call fee management service to update the fee as deferred. </p> @param applicationContext @param partyBundleLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	chargeBundleFee	This method will charge the bundle subscription fee and raise accounting when the nominated account is a loan account. @param accountId @param eventId @param priceDefinitionId @param priceDefinitionCurrency @param priceDefintionName @param feeAmount @param pricingTicketNumber @param deductionModeType @param narration @param baseNetFeesAmount @param additionalFee @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	This methods does auto disbursement As per data provided in LoanAutoDisbursementData. @param applicationContext @param accountId @param disbursementData @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	This method is responsible for fetching fees for an event Fees are defined in pricing module for a particular event. @param sessionContext @param accountId @param eventId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	This methods fetches the fee at Reco mode in origination. @param sessionContext @param accountId @param eventId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEvent	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	This method validate the execution date. @param branchCode @param exectionDate @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEventStrictlyAtReco	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	getPriceOnEventStrictlyAtReco	@param applicationContext @param accountId @param eventId @param transactionAmount @return @throws FatalException	This methods fetches the fee at Reco mode in origination. @param sessionContext @param accountId @param eventId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchDisbursementCount	<p>This method will fetch the loan account disbursement count. </p> @param accountId @return	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchDisbursementCount	<p>This method will fetch the loan account disbursement count. </p> @param accountId @return	This method will check if fee processing is to be done. If the nominated account is CASA or EXTERNAL, this method returns true. However, if the nominated account is LOANS, this method will check if the LOAN account has been disbursed. If disbursed, this method return true, else it returns false. @param accountId @param accountModule @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchDisbursementCount	<p>This method will fetch the loan account disbursement count. </p> @param accountId @return	This method will check if the input account number is in fixed rate of interest or variable. @param accountId @param accountModule @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountDuringInquiry		This method will validate the account id on the basis of the account Id. @param accountId @param accountType
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountDuringInquiry		The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	addLoanHardshipDetails		<p>This represents the addHardshipReliefRequest method of the entity. 1)Following steps are carried out in the method: 1)System Constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplcation is called via AccountLoanApdater, LoanHardhsipRelief whose approval type are "Pending" are returned from the called method. 5)If DDAHardsipReliefRequestDetails are present, then DDAHardsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.app.account.dto.hardshiprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipDetails		<p>This represents the updateHardshipReliefRequest method of the entity. Following steps are carried out in the method: 1)System constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplcation is called via AccountLoanApdater. 5)If DDAHardhsipReliefRequestDetails are present, then DDAHardhsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipDetails		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanHardshipReliefDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanHardshipReliefDetails		<p>This represents the fetchHardshipReliefRequest method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.app.account.hardshiprelief.HardshipReliefRequestInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanHardshipReliefDetails		<p>This method will apply the authorisation decision for hardship relief request. If the decision is Approved, request and reliefs would be marked APPROVED and STP reliefs will be applied. If the decision is Rejected, request and reliefs will be marked REJECTED. Whereas for decision being Returned, the request and reliefs will be marked RETURNED. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,hardshipApplicationId, ,hardshipApplicationId @fcb.param MI,HardshipReliefRequestStatus,hardshipReliefRequestStatus, ,hardshipReliefRequestStatus @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipDetails		<p>This represents the updateHardshipReliefRequest method of the entity. Following steps are carried out in the method: 1)System constraint are checked. 2)Business Policy are validated 3)If above steps are successful, hardship relief entity is added to the table. 4)If LoanHardshipReliefDetails are present, then LoanHardhsipReliefApplcation is called via AccountLoanApdater. 5)If DDAHardhsipReliefRequestDetails are present, then DDAHardhsipReliefApplication is called via AccountDAApdater. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipDetails		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	modifyTitle		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	modifyTitle		<p>This method is called by the public method modifyAccountTitles().&lt;br&gt;This method works as follows:&lt;br&gt; &lt;li&gt;Validate the party by calling {@link PartyApplicationService#fetchPartyDetails} by passing titleModificationRequest.partyId to it. &lt;li&gt;Create a hashmap called accountTitleMap. &lt;li&gt;Construct an string array of account ids by looping through the titleModificationRequest.titleModifications and adding each {@TitleModification#accountId} to it. At the same time, add an element to the accountTitleMap. Key : accountId. Value: newTitle. &lt;li&gt;Call {@link PartyAccountRelationshipApplicationService#fetchTypesForAccounts}. This method will return an array of {@link AccountTypeDTO} indicating the the type of each account that was sent. Type : CASA, TD , LOAN. &lt;li&gt;Create an arraylist of strings to contain all the accounts for which modification was successful. &lt;li&gt;Loop through the array of {@link AccountTypeDTO} and based on the type call the appropriate service to modify the title. Retrieve the title from accountTitleMap based on the accountId. switch(accountType) CASA: Call {@link DemandDepositAccountApplicationService#modifyAccountTitle} TD : Call {@link TermDepositAccountApplicationService#modifyAccountTitle} LOAN: TODO If no exception is thrown, then add the accountId to the arraylist. &lt;li&gt;Construct the response object {@link TitleModificationResponseDTO}. Add the array of account ids by calling toArray() on the arraylist. &lt;li&gt;Return @param sessionContext @param titleModificationRequest @param response @param transactionStatus @param tilModificationResponseDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchNoOfInstallmentsBetweenDates	This method is called to fetch number of installments between dates @param accountId @param reliefStartDate @param reliefExpiryDate @return	This method is used to fetch no of repayments between hardship start date and expiry date. @param sessionContext @param accountId @param reliefStartDate @param reliefExpiryDate @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanDetailsForEconomicCost	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> &lt;li>&lt;li>FacilityId&lt;/li>&lt;/li>&lt;li>&lt;li>OutstandingAmount&lt;/li>&lt;/li>&lt;li>&lt;li>CurrentFixedRate&lt;/li>&lt;/li>&lt;li>&lt;li>QuoteExpiryDate&lt;/li>&lt;/li>&lt;/pre> @param accountId @return EconomicCostQuoteDTO	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> &lt;li>&lt;li>FacilityId&lt;/li>&lt;/li>&lt;li>&lt;li>OutstandingAmount&lt;/li>&lt;/li>&lt;li>&lt;li>CurrentFixedRate&lt;/li>&lt;/li>&lt;li>&lt;li>QuoteExpiryDate&lt;/li>&lt;/li>&lt;/pre> @param accountId @return EconomicCostQuoteDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanDetailsForEconomicCost	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> &lt;li>&lt;li>FacilityId&lt;/li>&lt;/li>&lt;li>&lt;li>OutstandingAmount&lt;/li>&lt;/li>&lt;li>&lt;li>CurrentFixedRate&lt;/li>&lt;/li>&lt;li>&lt;li>QuoteExpiryDate&lt;/li>&lt;/li>&lt;/pre> @param accountId @return EconomicCostQuoteDTO	This method returns the following parameters required for Economic Cost Quote generation for given accountId and current date:-(This method to be called while generating the economic cost quote from the screen) <pre> <li>FacilityId</li><li>OutstandingAmount</li><li>CurrentFixedRate</li><li>QuoteExpiryDate</li></pre> @param accountId @return EconomicCostQuoteDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountPreferences	This method returns the Loan account preferences details for given accountId. @param accountId @return	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountPreferences	This method returns the Loan account preferences details for given accountId. @param accountId @return	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountPreferences	This method returns the Loan account preferences details for given accountId. @param accountId @return	This method returns the Loan account preferences details for given accountId. @param accountId @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountPreferences	This method returns the Loan account preferences details for given accountId. @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	isExistingAccount	This method is responsible for verifying existence of Loan Account @param sessionContext @param accountId @throws FatalException	System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link DDStatementRequestDTO},ddStatementRequestDTO , It contains all the information required to place a

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key : contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : StatementPeriodType MI. Denotes the period type of the input.</li> <li>period : Period MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : Date MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : Date must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing : FeeDetailsDTO contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType : StatementRequestType, MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus : StatementRequestStatus must be null. It will be set to pending.</li> <li>statementNo : Must be set to null for statement order. It will be set at statement generation.</li> <li>customerInitiatedRequest : CustomerInitiatedRequest Must be null.</li> </ul> <p>@return Response. It contains the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link DDStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	isExistingAccount	This method is responsible for verifying existence of Loan Account @param sessionContext @param accountId @throws FatalException	System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order term deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,{@link TDStatementRequestDTO},tdStatementRequestDTO , , It contains all the information required to place a demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>next working day.</p> <ul style="list-style-type: none"> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.</li> <li>statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.</li> <li>statementNo : Must be set to null for statement order.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.</li> </ul> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDueDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt;  &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.&lt;/li&gt;  &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null. Will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;  &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null&lt;/li&gt; &lt;/ol&gt;  @return {@link TDStatementRequestInquiryResponse}. It contains the {@link TDStatmentRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;  &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	isExistingAccount	This method is responsible for verifying existence of Loan Account @param sessionContext @param accountId @throws FatalException	System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI, {@link LoanStatementRequestDTO},loanStatementRequestDTO, , It contains all the information required to place a

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>demand deposit statement request. &lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt; &lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>customer initiated request details.</p> <p>For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>actionDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.</li> <li>statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.</li> <li>statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending</li> <li>statementNo : Must be set to null for statement order.It will be set at statement generation.</li> <li>customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null</li> </ul> <p>@return {@link LoanStatementRequestInquiryResponse}. It contains</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>the {@link LoanStatementRequestDTO} of the demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;</p> <p>&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt;</p> <p>&lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt;</p> <p>&lt;ul&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	<p>This method provides financial and non financial transaction details for input filter provided on term deposit account. On financial or non financial transaction event, application stores transaction details in terms of <a href="#">StatementItem</a>.</p> <p>The list of the financial or non financial events on which details are logged is fixed. Financial transaction generally updates balances of the account for example TD payin, redemption, Interest settlement etc. Non financial transaction does not update balances but they update non financial attributes of the account account address. Transaction inquiry request may come from branch banking or direct banking.</p> <p>In case of branch banking operation executive may want to see transaction performed on particular account within specified period.</p> <p>In case of direct banking ,customer may want to see transactions performed on account. <a href="#">TransactionInquiryDTO</a> contains start and end date .This method lists all the transactions where <a href="#">TransactionAmount#transactionDate</a> is greater or equal than <a href="#">TransactionInquiryDTO#startDate</a> and <a href="#">TransactionAmount#transactionDate</a> is less than or equal to <a href="#">TransactionInquiryDTO#endDate</a> . Please refer <a href="#">StatementInquiryFilterInputDTO</a> for more details .</p> <p>Financial transaction details contains following details.</p> <ul style="list-style-type: none"> <li>Transaction Amounts: Amount in different currencies.</li> <li>Transaction key/ Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction.</li> <li>Service Charge Details: Pricing reference number or pricing ticket</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>from which details can be asked from pricing module.</p> <ul style="list-style-type: none"> <li>Transaction source details- channel and application user.</li> <li>Transaction dates: Different dates indicating realization of transaction.</li> <li>Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic.</li> </ul> <p>Non financial transaction contain following details</p> <ul style="list-style-type: none"> <li>Event date : Date on which event is performed.</li> <li>Event description : description of the event like "Interest rate is changed from 5% to 4.5%"</li> </ul> <p>@param transactionInquiryDTO MI, TransactionInquiryDTO, transactionInquiryDTO, It contains the following attributes:</p> <ul style="list-style-type: none"> <li>accountId: String MI it is the id of the account for which TD transactions are inquired for.</li> <li>statementPeriodType: StatementPeriodType MI denotes the period type for which the inquiry is done</li> <li>statementInquiryFilterInputDTO: StatementInquiryFilterInputDTO, it contains the following attributes: <ul style="list-style-type: none"> <li>statementInquiryFilterTypes: StatementInquiryFilterType[] it has all the filters that need to be applied.</li> <li>Optional startAmount: MoneyDTO MI if amount range filter is chosen. Denotes the start amount range.</li> <li>endAmount: MoneyDTO MI if amount range filter is chosen. Denotes the end amount range.</li> <li>checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.</li> <li>checkNumberEnd: String MI if cheque range filter is chosen. Denotes the end range.</li> <li>transactionDescription: String MI if transaction description filter is chosen. Denotes the</li> </ul> </li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>description for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;transactionMnemonics: String MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;lastNumberOfTransactions: int. Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs. &lt;li&gt;lastNumberOfDays: int. This denotes the last number of days for which the transactions are to be fetched. If this is set to zero, then default values will be picked up from statement parameters. &lt;/ol&gt; &lt;/li&gt; &lt;li&gt;startDate :{@link Date} Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods. Also if either of Last_X_Transaction or Last_N_Days filter are used, it can be null. It is a mandatory input for all other cases.&lt;/li&gt; &lt;li&gt;endDate : {@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.&lt;/li&gt; &lt;li&gt;isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.&lt;/li&gt; &lt;/ol&gt; @return {@link LoanTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link LoanStatementItemDTO}.Each LoanStatementItemDTO can be of following type.  &lt;ol&gt; &lt;li&gt; {@link LoanFinancialStatementItemDTO}: Loan financial Item.- Contains financial details.  &lt;/ol&gt; All the different types of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanStatementItemDTO are combined and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.</p> <p>The following sections indicate the completion of the transactions with reasons:</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</a></li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</a></li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li><a href="#">{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_FILTER_CANNOT_BE_USED_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_AMOUNT_GREATER_THAN_END_AMOUNT}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_CHEQUE_NUMBER_START_GREATER_THAN_CHEQUE_NUMBER_END}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_FILTER_MISMATCH_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}</a></li> </ol> <p>@throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	<p>Customers may want to see the transactions that have occurred in their loan account. For this they will have to submit a statement request. Every statement request that a customer makes is tracked as <a href="#">{@link LoanStatementRequest}</a> for loan accounts. The customer may also want to reorder the same statement request as before. To enable this facility, this method will fetch all the statement request that have been generated in the given period by system. From this, customer can then reorder any one, multiple or all of the requests present for his particular account. The method will not fetch statement request which are just reorders.</p> <p>Consider the following example:</p> <p>Suppose there is a statement request for the period 'x' to 'y' with statement id 'z'. Now if the customer reorders this request, there will be one more statement request generated with 'z' as the reorder id. Lets say the new statement request id is 'newZ'. This statement id will not be fetched by this service. The same is true even if the reorder is bulk.</p> <p>@param MI,SessionContext,sessionContext, This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes.</p> <p>@param MI,String,accountId, This parameter denotes the casa accountId for which the statement requests are to be fetched.</p> <p>@param MI,Period,period, This denotes the date range for which the statement requests are to be fetched.</p> <p>It contains:</p> <ul style="list-style-type: none"> <li>beginDate: MI;{@link Date}. It denotes the start date from which the statement requests are to be fetched.It must not be after the endDate.</li> <li>endDate: MI;{@link Date}. It denotes the end date to which the statement requests are to be fetched. It must not be before the start date.</li> </ul> <p>@return {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>LoanStatementRequestArrayInquiryResponse} This response contains the system as well as customer initiated requests {@link LoanStatementRequest} for the given period that are system initiated, assembled into {@link LoanStatementRequestDTO}. It also contains the Transaction Status. The following sections indicate the completion of the transactions with reasons</p> <ul style="list-style-type: none"> <li><b>Transaction Success</b></li> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS</a></li> <li><b>Transaction Failure</b></li> <li><a href="#">com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR</a></li> </ul> <p>Error Reason may be one of the following</p> <ul style="list-style-type: none"> <li><a href="#">AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR</a></li> <li><a href="#">AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE</a></li> <li><a href="#">AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR</a></li> <li><a href="#">AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR</a></li> <li><a href="#">AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR</a></li> </ul> <p>@throws FatalException</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method fetches the account details of the given Loan Account Id. @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method will return the primary account holder of the account, @param sessionContext @param accountId @param accountType @return String
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method will check the account preference for validating the channel facility for that particular account @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method fetches the account details of the given Loan Account Id. @param accountId @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This methods does auto disbursement As per data provided in LoanAutoDisbursementData. @param applicationContext @param accountId @param disbursementData @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method does the Subsequent Disbursement for the loan Account. @param sessionContext @param accountId @param scheduleTypeCode @param definitionDate @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	@param applicationContext @param disbursementData @param repaymentStages @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method validates the host disbursement by applying business policies. @param accountId @param disbursementAmount @param disbursementDate
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method implements the process method for Auto Disbursement Action on the account. @param context @param accountId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method do the Auto Disbursement For SplitLoan @param applicationContext @param accountId @param splitLoanAccountInstruction @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method performs Disbursement posting considering the calculated fees.  It performs following tasks:-   <ul> <li>Inquire Settlement Details</li> <li>Prepare Fee Collection Details</li> <li>Post the Disbursement using Disbursement Posting Application</li> </ul> @param applicationContext @param deductionModeType @param eventId @param accountId @param disbursementType @param disbursementCount @param loanAutoDisbursementData @param wrappedPriceFeatureDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This methods update the Contribution Received towards loan. @param accountId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method returns the Loan account preferences details for given accountId. @param accountId @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method is responsible for verifying existence of Loan Account @param sessionContext @param accountId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method is responsible for fetching the account. @param accountId @return AccountDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method fetches the pricing ticket no for an arrangement Id.
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method reverses the backdated disbursement done for a loan account. If first disbursement is to be reversed then reverseDisbursement is called. In case of subsequent disbursement, reverse method of LoanTransactionReversalApplicationService is called. @param sessionContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method is used to close a loan account. <li>Reference number is set into the transaction details DTO and the loan account is loaded.</li> <li>Instances of LoanFullPayOffSystemConstraintDTO and LoanFullPayOffSystemConstraintDTO are created.</li> <li>System Constraints are checked and Economic cost for the account is fetched.</li> <li>The instance of LoanFullPayOffBusinessPolicyFactory is created and then the policy is validated.</li> <li>Online posting is done with the help of processNormalTransaction method.</li> <li>The reason for closure is updated.</li> Then the activity is logged for alerts. @param sessionContext @param eventId @param accountId @param transactionDetailsDTO @param componentPreferenceDTO @param wrappedPriceFeatureDTOs @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method will be responsible for posting the normal transaction all along.  @param applicationContext @param accountId @param eventId @param transactionDetailsDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method will prepare the {@link #EventData)} out of ApplicationContext, loanAccount and TransactionDetails DTO. @param applicationContext @param accountId @param eventId @param transactionDetailsDTO @return EventData
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccount	This method is responsible for fetching the account. @param accountId @return AccountDTO	This method sets the call the settlement module to fetch the component preference Id.Settlement payout instruction,contribution payin instruction and disbursement instruction will individually call settlement module for get the component preference Id. This settlement reference Id will send to module.Calling modules must keep a track of the id at the account level if they wish to execute this preference later. @param sessionContext @param loanAdapter @param settlementPayoutInstructionDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processFeeRefundTransaction	This method will process the reverse fees which will be refunded to the corresponding CASA account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param accountNo @param reversalDetailsDTO @throws FatalException	This method will process the reverse fees which will be refunded to the corresponding CASA/LOAN account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param sessionContext @param accountNo @param reversalDetailsDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	modifyStatementAttributes	<p>This method is responsible for modifying Statement related parameters, which are being maintained at Loan Account.</p> <p>Following parameters are maintained:-</p> <ul style="list-style-type: none"> <li>Last Statement Opening Balance.</li> <li>Last Statement Closing Balance.</li> <li>Date of Last Statement Generation.</li> </ul> <p>Parameters: @param accountId @param stmtOpeningBalance @param stmtClosingBalance @param period</p>	<p>This method is responsible for processing System Initiated Request for Loan Statement Generation.</p> <p>Parameters: @param currentStatementRequest @param demandDepositAccountDTO @param statementparametersDTO</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	modifyStatementAttributes	<p>This method is responsible for modifying Statement related parameters, which are being maintained at Loan Account.</p> <p>Following parameters are maintained:-</p> <ul style="list-style-type: none"> <li>Last Statement Opening Balance.</li> <li>Last Statement Closing Balance.</li> <li>Date of Last Statement Generation.</li> </ul> <p>Parameters: @param accountId @param stmtOpeningBalance @param stmtClosingBalance @param period</p>	<p>This method is responsible for modifying Statement related parameters, which are being maintained at Loan Account.</p> <p>Following parameters are maintained:-</p> <ul style="list-style-type: none"> <li>Last Statement Opening Balance.</li> <li>Last Statement Closing Balance.</li> <li>Date of Last Statement Generation.</li> </ul> <p>Parameters: @param accountId @param stmtOpeningBalance @param stmtClosingBalance @param period</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanAccountConfiguration	<p>This method is responsible for inquiring the Account configuration done in Origination Systems for the arrangement Id.</p> <p>Parameters: @param sessionContext @param arrangementId @return @throws FatalException</p>	<p>This method fetches the pricing ticket no for an arrangement Id.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseDisbursement	This method is responsible for reversal of disbursement transaction carried out as part of settlement payout transaction. @param sessionContext @return transactionStatus	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseDisbursement	This method is responsible for reversal of disbursement transaction carried out as part of settlement payout transaction. @param sessionContext @return transactionStatus	This method reverses the disbursement done for a loan account. If first disbursement is to be reversed then reverseDisbursement is called. In case of subsequent disbursement, reverse method of LoanTransactionReversalApplicationService is called. @param sessionContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseDisbursement	This method is responsible for reversal of disbursement transaction carried out as part of settlement payout transaction. @param sessionContext @return transactionStatus	This method reverses the backdated disbursement done for a loan account. If first disbursement is to be reversed then reverseDisbursement is called. In case of subsequent disbursement, reverse method of LoanTransactionReversalApplicationService is called. @param sessionContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseDisbursement	This method is responsible for reversal of disbursement transaction carried out as part of settlement payout transaction. @param sessionContext @return transactionStatus	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseBackdatedDisbursement	This method reverses the backdated disbursement done for a loan account. If first disbursement is to be reversed then reverseDisbursement is called. In case of subsequent disbursement, reverse method of LoanTransactionReversalApplicationService is called. @param sessionContext @param instructionId	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseContribution	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseContribution	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId	This method will reverse the contribution instruction for the settlement payout instruction.  It will interact with Loan module for the reversal of the contribution instruction via  AccountLoanAdapter.  @param applicationContext @param instructionId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseContribution	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseContribution	This method will reverse the contribution instruction for the settlement payout instruction.  @param applicationContext @param instructionId	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipRelief	This method is used to individual update of hardship relief status. @param applicationContext @param requestId @param hardShipReliefStatusType @param reasonForRejection @throws FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipRelief	This method is used to individual update of hardship relief status. @param applicationContext @param requestId @param hardShipReliefStatusType @param reasonForRejection @throws	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanHardshipRelief	This method is used to individual update of hardship relief status. @param applicationContext @param requestId @param hardShipReliefStatusType @param reasonForRejection @throws FatalException	@param sessionContext @param requestId @param hardshipAccountType @param reliefStatusType @param reasonForRejection @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This represents the updateAccountCreditMatrix method of the entity.
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This represents the closeAccountCreditMatrix method of the entity.
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This represents the reopenAccountCreditMatrix method of the entity.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		<p>This method modifies the address linkages for all the parties associated with the account. Steps to be performed are 1) Set the task code of the operation for Interaction 2) Create Transaction Context by passing SessionContext and ADDITION (as MaintenanceType) as an arguments 3) Based on the account type, check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid, "Invalid Account Number" error message is displayed. If the account type is CASA, call fetchBasicDetails() method exposed by the DDInquiryApplicationService with accountId as the input. If the account type is TD, call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with accountId as the input. 4) Call validateAccount() method with accountId and status as the input. Only accounts with status "regular" are allowed to have linkages maintained. If the account status is closed or dormant, "Closed/Dormant Account" error message would be displayed. 5) Check for the validity of the each party and its account relationship by calling fetchPartyAccountRelationship() method exposed by the PartyAccountRelationshipApplicationService. If the party is not valid, "Invalid Party" error message is displayed. If the party does not have any relationship with the account, "No Relationship with the Account" error message is displayed. 6) i) Call validateCombinedStatementSelection() method with AccountAddressLinkageDTO as the input. If the validation fails, error message would be displayed. ii) Call validateHoldOption() method with AccountAddressLinkageDTO as the input. If the validation fails, error message would be displayed. 7)</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Create an object of AccountAddressLinkageAssembler. Convert the AccountAddressLinkageDTO to domain object using the AccountAddressLinkageAssembler. 8) Create an instance of Domain Service (AccountAddressLinkageService). 9) Call the modifyAccountAddressLinkage() method of the AccountAddressLinkageService with the instance of AccountAddressLinkage created in the above step. 10) Fill the Transaction Status by calling the method fillTransactionStatus. 11) Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks. 12) Close the interaction in finally block. 13) Return the transactionStatus object as response of this operation. @param accountAddressLinkageDTO @param sessionContext
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This method does the Subsequent Disbursement for the loan Account. @param sessionContext @param accountId @param scheduleTypeCode @param definitionDate @param disbursementAmount @param feeCollectionDTO @param componentPreferenceDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This method implements the process method for Auto Disbursement Action on the account. @param context @param accountId @param actionDate @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		This method is called to fetch the collections details for a loan account. @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		<pre> This method will process the Loan Consolidation which includes&amp;lt;br&amp;gt; 1 - Validating the Loan Consolidation.&amp;lt;br&amp;gt; 2 - Closing all the child accounts.&amp;lt;br&amp;gt; 3 - Doing Sanction Amount Revision of the parent account by an Amount equal to the sum of expected payOff's of all the child accounts.&amp;lt;br&amp;gt; 4 - Doing subsequent Disbursement of the parent account. &amp;lt;br&amp;gt; 5 - Apply economic cost on the parent account, if applicable. &amp;lt;br&amp;gt; @param sessionContext @param loanConsolidationInputDTO @param feeCollectionDTO @return TransactionStatus @throws FatalException </pre>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchCollectionDetails		<pre> This method will be responsible for posting the normal transaction all along.&lt;br&gt; @param applicationContext @param accountId @param eventId @param transactionDetailsDTO @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollectionOnAddressTypeChange	This method would intimate collection on address type change for default comm address @param accountId @param defaultCommAddressType	This method modifies the address linkages for all the parties associated with the account. Steps to be performed are 1) Set the task code of the operation for Interaction 2)Create Transaction Context by passing SessionContext and ADDITION (as MaintenanceType) as an arguments 3) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDInquiryApplicationService with accountId as the input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with accountId as the input. 4) Call validateAccount() method with accountId and status as the input. Only accounts with status " regular" are allowed to have linkages maintained. If the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 5) Check for the validity of the each party and its account relationship by calling fetchPartyAccountRelationship() method exposed by the PartyAccountRelationshipApplicationService. If the party is not valid, "Invalid Party" error message is displayed. If the party does not have any relationship with the account, "No Relationship with the Account" error message is displayed. 6) i) Call validateCombinedStatementSelection() method with AccountAddressLinkageDTO as the input. If the validation fails, error message would be displayed. ii) Call validateHoldOption() method with AccountAddressLinkageDTO as the input.If the validation fails, error message would be displayed. 7)

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Create an object of AccountAddressLinkageAssembler. Convert the AccountAddressLinkageDTO to domain object using the AccountAddressLinkageAssembler. 8) Create an instance of Domain Service (AccountAddressLinkageService). 9) Call the modifyAccountAddressLinkage() method of the AccountAddressLinkageService with the instance of AccountAddressLinkage created in the above step. 10) Fill the Transaction Status by calling the method fillTransactionStatus. 11) Put the above set of operations in try-catch block and specify the possible exceptions in catch blocks. 12) Close the interaction in finally block. 13) Return the transactionStatus object as response of this operation. @param accountAddressLinkageDTO @param sessionContext
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseLMIAAndLegalFeeArrears	@param nominatedAccount @param lmiPremium @param legalFee	This method is used to @param applicationContext @param instructionId @param actionDate
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchStatementFrequencyForLoanAccount		<p>This method is responsible for placing Next System Initiated Request for Loan Statement Generation. </p> @param accountId @param loanAccountDTO @param statementparametersDTO @param statementGenTill
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	reverseRaisedArrears	This method would reverse all the raised arrears on the current value date. @param applicationContext @param accountId @param originalTransactionReferenceNo @param deductionModeType	This method would reverse all the raised arrears on the current value date. @param applicationContext @param accountId @param originalTransactionReferenceNo @param deductionModeType

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseReverseAccountingEvent	This method is used to raise reverse accounting event based on originalTransactionReferenceNumber @param applicationContext @param originalTransactionReferenceNumber @param accountEventCode	This method is used to raise reverse accounting event based on originalTransactionReferenceNumber @param applicationContext @param originalTransactionReferenceNumber @param accountEventCode
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseReverseAccountingEvent	This method is used to raise reverse accounting event based on originalTransactionReferenceNumber @param applicationContext @param originalTransactionReferenceNumber @param accountEventCode	This method does the reversal of settlement payout transaction. @param sessionContext @param settlementPayoutSetupId @return transactionStatus @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseReverseAccountingEvent	This method is used to raise reverse accounting event based on originalTransactionReferenceNumber @param applicationContext @param originalTransactionReferenceNumber @param accountEventCode	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	applyLoanHardshipDetails		This method will mark the hardship reliefs and request APPROVED and apply the STP reliefs. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateOffsetAccountFlag	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException	This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateOffsetAccountFlag	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException	This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateOffsetAccountFlag	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException	This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateOffsetAccountFlag	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException	This represents the deleteOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateOffsetAccountFlag	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException	This method updates the offset account flag of the account. @param sessionContext @param accountId @param hasOffsetAccount @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logNonFinancialStatementOnOffsetLinkage	This method is responsible for logging Non Financial Statement when CASA Account would be linked to Loan Account as "Offset Account". @param applicationContext @param offsetAccountDTO	This method is responsible for logging Non Financial Statement when CASA Account would be linked to Loan Account as "Offset Account". @param applicationContext @param offsetAccountDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logNonFinancialStatementOnOffsetDeLinkage	This method is responsible for logging Non Financial Statement when CASA Account would be delinked from Loan Account as "Offset Account". @param applicationContext @param offsetAccountDTO	This method is responsible for logging Non Financial Statement when CASA Account would be linked to Loan Account as "Offset Account". @param applicationContext @param offsetAccountDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatus		<p>This method is responsible for processing System Initiated Request for Loan Statement Generation. </p> @param currentStatementRequest @param demandDepositAccountDTO @param statementparametersDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatus		<p>This method is used to validate Loan Account status.   If the System defined or User defined status of the current account has been marked as blocked at statement parameter level,   then generation process will be halted. It returns the built up error message for each blocked status that is encountered.  For closed accounts, the generation process will be done only if the closure date falls in the period from the current statement period begin date till the end date plus deferred statement generation days if any. In this case it will be assumed to be the final statement request. </p> @param currentStatementRequest @param accountId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatus		This method is to check the account Status @param applicationContext @param instructionId @param actionDate @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatus		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatus		This method validates whether auto disbursement can be done on the account by applying the business policies @param context @param disbursementData @return boolean @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatusForSettlement		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateAccountStatusForSettlement		This method validates whether auto disbursement can be done on the account by applying the business policies @param context @param disbursementData @return boolean @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchPreBalanceDetails		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchPreBalanceDetails		This method is used to perform backdated subsequent disbursement. @param applicationContext @param loanAccount @param eventId @param deductionModeType @param wrappedPriceFeatureDTO @param disbursementData @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchPreBalanceDetails		<pre> This method performs the doSubsequentDisbursement operation. It performs the following steps: &lt;ol&gt; &lt;li&gt;doRescheduleOnSubsequentDisbursement&lt;/li&gt; &lt;li&gt;validateSchedule&lt;/li&gt; &lt;li&gt;generateRepaymentRecordsOnReschedule&lt;/li&gt; &lt;li&gt;saveScheduleOnReschedule&lt;/li&gt; &lt;li&gt;post the Disbursement Transaction&lt;/li&gt; &lt;li&gt;raise Accounting&lt;/li&gt; &lt;/ol&gt; @param applicationContext @param accountId @param savedScheduleArrangementId @param disbursementDate @param topupDisbursementAmount @param capitalizeArrears @param feeCollectionDTO @param eventData @param amountTagToDirectionMap @throws FatalException </pre>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchPreBalanceDetails		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchPreBalanceDetails		<pre> This method will process the Loan Consolidation which includes: &lt;br&gt; 1 - Validating the Loan Consolidation. &lt;br&gt; 2 - Closing all the child accounts. &lt;br&gt; 3 - Doing Sanction Amount Revision of the parent account by an Amount equal to the sum of expected payOff's of all the child accounts. &lt;br&gt; 4 - Doing subsequent Disbursement of the parent account. &lt;br&gt; 5 - Apply economic cost on the parent account, if applicable. @param sessionContext @param loanConsolidationInputDTO @param feeCollectionDTO @return </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				TransactionStatus @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateAccountUtilization		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateAccountUtilization		This method is used to perform backdated subsequent disbursement. @param applicationContext @param loanAccount @param eventId @param deductionModeType @param wrappedPriceFeatureDTO @param disbursementData @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateAccountUtilization		<pre>This method performs the doSubsequentDisbursement operation. It performs the following steps: &lt;ol&gt; &lt;li&gt;doRescheduleOnSubsequentDisbursement&lt;/li&gt; &lt;li&gt;validateSchedule&lt;/li&gt; &lt;li&gt;generateRepaymentRecordsOnReschedule&lt;/li&gt; &lt;li&gt;saveScheduleOnReschedule&lt;/li&gt; &lt;li&gt;post the Disbursement Transaction&lt;/li&gt; &lt;li&gt;raise Accounting&lt;/li&gt; &lt;/ol&gt; &#064;param applicationContext &#064;param accountId &#064;param savedScheduleArrangementId &#064;param disbursementDate &#064;param topupDisbursementAmount &#064;param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				capitalizeArrears @param feeCollectionDTO @param eventData @param amountTagToDirectionMap @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateAccountUtilization		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateAccountUtilization		<pre>This method will process the Loan Consolidation which includes  1 - Validating the Loan Consolidation.  2 - Closing all the child accounts.  3 - Doing Sanction Amount Revision of the parent account by an Amount equal to the sum of expected payOff's of all the child accounts.  4 - Doing subsequent Disbursement of the parent account.   5 - Apply economic cost on the parent account, if applicable.   @param sessionContext @param loanConsolidationInputDTO @param feeCollectionDTO @return TransactionStatus @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollectionOnAccountCreditMatrix	This method would intimate collection on account credit matrix updates/close/re-open @param accountId @param feederDTO	This method would intimate collections in case a account belongs to Loans. @param accountDDAAdapter @param accountCreditMatrixDTO
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	levyOnlineFees		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	levyOnlineFees		<pre>This method will process the Loan Consolidation which includes  1 - Validating the Loan Consolidation.  2 - Closing all the child accounts.  3 - Doing Sanction Amount Revision of the parent account by an Amount equal to the sum of expected payOff's of all the child accounts.  4 - Doing subsequent Disbursement of the parent account.   5 - Apply economic cost on the parent account, if applicable.   &#064;param sessionContext &#064;param loanConsolidationInputDTO &#064;param feeCollectionDTO &#064;return TransactionStatus @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processPeriodicFees		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processPeriodicFees		
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		This method is used to process the subscription & maintenance fees during BOD batch process. @param context @param valueDateBatchData
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		<p>This method will charge the inquired fees. </p> @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param periodicPriceFeatureDTOArray @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		<p>&lt;p&gt; Method does maintenance fee processing for account modules &lt;/p&gt; &lt;p&gt; It fetches the account basic details having information of party, currency etc. &lt;/p&gt; &lt;p&gt; For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data &lt;/p&gt; @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		<p>This method is used to process the subscription &amp; maintenance fees during batch process. @param context @param valueDateBatchData</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		<p>&lt;p&gt;This method will charge the inquired fees. &lt;/p&gt; @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param periodicPriceFeatureDTOArray @throws FatalException</p>
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	processBundleFees		<p>&lt;p&gt; Method does maintenance fee processing for account modules &lt;/p&gt; &lt;p&gt; It fetches the account basic details having information of party, currency etc. &lt;/p&gt; &lt;p&gt; For loan account it determines if first disbursement has happened. If so then it will raise arrear for loan account for maintenance fees else will log into audit data &lt;/p&gt; @param applicationContext @param bundleFeeBatchData @param partyBundleLinkage @param nominatedAccountModule @param periodicPriceFeatureDTOArray @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is minor and the operating instruction channel access type is no for given channel then return true,false otherwise @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will fetch the bank code for given account ID @param partyAccountRelationshipDetailsDTO @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchBankCode	This method will fetch the bank code for given account code @param accountId @return	This method will check if the party is operationally major and the operating instruction channel access type is no for given channel then return true,false otherwise @param partyAccountRelationshipDetailsDTO @return
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	addLoanRolloverInstruction	This method calls the loan module to add a loan roll over instruction. @param sessionContext the {@link SessionContext} @param loanRolloverInstructionDTO the {@link LoanRolloverInstructionDTO} @param rateLockDTO the {@link RateLockDTO} @param feeCollectionDTO the {@link FeeCollectionDTO} @return the {@link TransactionStatus} @throws FatalException	This method will confirm an Account Variation Request.  It will first confirm the proposed facilities for the variation request  It will then do the variation request transaction depending the variation request type. For eg., for {@link VariationRequestType#LOAN_ROLLOVER_REQUEST}, it will call the Loan module's {@link IAccountLoanAdapter#addLoanRolloverInstruction(SessionContext, com.ofss.fc.app.loan.dto.rolloverinstruction.LoanRolloverInstructionDTO, com.ofss.fc.app.loan.dto.ratelock.RateLockDTO, com.ofss.fc.app.loan.dto.feeCollectionDTO)} .  It will then mark the variation request status as completed. @param sessionContext the {@link SessionContext} @param variationRequestId the Variation Request Id @param facilityId the Facility Id @return the {@link TransactionStatus} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanDisbursementInstruction	@param sessionContext @param accountId @return @throws FatalException	@param sessionContext @param accountId @return @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchLoanDisbursementInstruction	@param sessionContext @param accountId @return @throws FatalException	This method will be called for fetching the component preference DTOs maintained for contribution, disbursement and settlement payout. Fetch method of settlement payout will internally call this method for fetching the componentPreferenceDTO maintained for all the instructions. This will then call inquireComponentPreferenceDetails() method for fetching the details for that instruction and account Id through account settlement adapter. @param sessionContext @param settlementPayoutInstructionDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	simulateHardshipRelief	The method is responsible to simulate hardship relief for a loan account. @param applicationContext @param loanHardshipDetailsDTOs @return @throws FatalException	The method is responsible to simulate hardship relief for a loan account. @param applicationContext @param loanHardshipDetailsDTOs @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	simulateHardshipRelief	The method is responsible to simulate hardship relief for a loan account. @param applicationContext @param loanHardshipDetailsDTOs @return @throws FatalException	The method is responsible to simulate Hardship relief request. The method will:  1 . Validate Hardship relief request.  2 . Apply hardship in Validate mode.  3 . Simulate hardship relief request. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,HardshipReliefRequestDTO,hardshipReliefRequestDTO, ,hardshipReliefRequestDTO TODO: Explain the hardshipReliefRequestDTO above @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	populateAccountScheduleLinkageDetailsInLoanAccountAggregate	This method populate the AccountScheduleLinkageDetails in LoanAccountAggregate. @param scheduleId @param accountId @param amountType	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchScheduleIdFromLoanAccountAggregate	This method fetch the scheduleId from LoanAccountAggregate @param accountId @param amountType @return	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	fetchScheduleIdFromLoanAccountAggregate	This method fetch the scheduleId from LoanAccountAggregate @param accountId @param amountType @return	This method fetch the scheduleId from LoanAccountAggregate @param accountId @param amountType @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateLoanDisbursementInstructionStatus	This method updates the status of Loan Disbursement Instruction @param disbursementInstructionKey	This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param applicationContext @param settlementPayoutInstruction
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateContributionInstructionStatus	This method updates the status of contribution Instruction @param accountId @param instructionId	This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param applicationContext @param settlementPayoutInstruction
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateContributionInstructionStatus	This method updates the status of contribution Instruction @param accountId @param instructionId	This method is used to update the {@link InstructionExecutionStatusType} of {@link SettlementPayoutInstruction}, {@link SettlementPayoutItem} and {@link DisbursementInstruction} to {@link InstructionExecutionStatusType#UNINITIATED} as part of same day reversal for the executed instructions. @param settlementPayoutInstruction
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateDelinquencyCountersOnWriteOff	This method has been defined to provide details of written of Arrears to update Delinquency Counters. @param writtenOfArrear	This service updates the arrear in the respect of arrear write off. @param applicationContext @param eventId @param arrearToWriteOff @param dueAmount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateDelinquencyCountersOnWriteOff	This method has been defined to provide details of written of Arrears to update Delinquency Counters. @param writtenOfArrear	This method is responsible for updating Delinquency Counters for the Arrears which have been written off completely.  Arrears which are being marked as "PAID" due to either partial / full write-off are chosen for Delinquency Counters update. @param arrearToWriteOff
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateDelinquencyCountersOnWriteOff	This method has been defined to provide details of written of Arrears to update Delinquency Counters. @param writtenOfArrear	This method has been defined to provide details of written of Arrears to update Delinquency Counters. @param writtenOfArrear
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollections	This method will update collections for any financial transaction on account if the account is in collections. @param accountId @param eventId	This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollections	This method will update collections for any financial transaction on account if the account is in collections. @param accountId @param eventId	This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollections	This method will update collections for any financial transaction on account if the account is in collections. @param accountId @param eventId	This represents the addMultipleOffsetAccounts method of the entity. @fc.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fc.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollections	This method will update collections for any financial transaction on account if the account is in collections. @param accountId @param eventId	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	updateCollections	This method will update collections for any financial transaction on account if the account is in collections. @param accountId @param eventId	The method will be invoked by hardship relief process after the decision for all the relief types have been taken. @param sessionContext @param partyId @param hardshipApplicationId @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logOffsetBenefitReversalNonFinancialStatementItem	This method will log the non-financial statement item for the offset benefit reversal. @param interestBenefitResponse	This method will reverse the offset benefit amounts having end date greater than input from date. This method eventually calls the InterestBenefitService to reverse the Offset Benefit Amounts.If isNewCharging is true which is in some cases of Loans, thenThis method is called. This method internally calls InterestBenefitService which deletes the interestBenefitLogItems present from the backdatedValueDate provided. The list of {@link InterestBenefitResponse} is sent from the InterestBenefitService which is iterated over and used for logging Non Financial Statement Item. @param accountId @param fromDate
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateLoanHardshipReliefDetails	This method will validate if an account is in MOR/UDI stage, if yes then hardship relief will not be allowed. @param applicationContext @param loanHardshipDetailsDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateLoanHardshipReliefDetails	This method will validate if an account is in MOR/UDI stage, if yes then hardship relief will not be allowed. @param applicationContext @param loanHardshipDetailsDTO	
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logActivityForBatch		This method is used for logging the details for raising the alert @param applicationContext @param offsetAccountDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logActivityForBatch		This method logs the activity for alert processing. For 1) ActivityId = "LN_DISBURSEMENT_BY_BATCH", we fetch the eventData. The details stored in the eventData are required for logging alerts. 2) ActivityId = "LN_RESCHEDULE_By_BATCH", we send eventData = null, as no additional details are required while logging the Alerts. The data for sending the alerts is directly fetched in ActivityEventHelper. @param applicationContext @param disbursementData @param activityId @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logActivityForBatch		This method is used to raise alert for offset account @param applicationContext @param offsetAccountDTO @param loanActivityId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logActivityForBatch		<pre>This method will process the Loan Consolidation which includes  1 - Validating the Loan Consolidation.  2 - Closing all the child accounts.  3 - Doing Sanction Amount Revision of the parent account by an Amount equal to the sum of expected payOff's of all the child accounts.  4 - Doing subsequent Disbursement of the parent account.   5 - Apply economic cost on the parent account, if applicable.   &#064;param sessionContext &#064;param loanConsolidationInputDTO &#064;param feeCollectionDTO &#064;return TransactionStatus @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestBenefitAction		This represents the addOffsetAccount method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO,offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestBenefitAction		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestBenefitAction		This represents the addMultipleOffsetAccounts method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,OffsetAccountDTO[],offsetAccountDTO, @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<i>Transaction Failure</i> section
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestBenefitAction		This method is used to log the interest benefit action. @param offsetAccountDTO @param offset
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	logInterestBenefitAction		This method is used to log interest benefit action. @param accountId
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateMaximumOffsetAccounts	This method will validate the maximum offset accounts that can be linked to the loan account with that defined at bank policy definition. @param sessionContext @param lengthOfOffsetAccountDTOs @throws FatalException	@param sessionContext @param offsetAccountDTO @throws FatalException
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	validateMaximumOffsetAccounts	This method will validate the maximum offset accounts that can be linked to the loan account with that defined at bank policy definition. @param sessionContext @param lengthOfOffsetAccountDTOs @throws FatalException	This method will validate the maximum offset accounts that can be linked to the loan account with that defined at bank policy definition.
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseReversalAccountingEventsForPayout		This method does the reversal of settlement payout transaction. @param sessionContext @param settlementPayoutSetupId @return transactionStatus @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAccountLoanAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\account\IAccountLoanAdapter	raiseReversalAccountingEventsForPayout		
ILogScannerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\logscanner\ILogScannerAdapter	generateLog		
ILogScannerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\logscanner\ILogScannerAdapter	generateLog		
ICollateralUpdateRealizationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\ICollateralUpdateRealizationAdapter	updateCollateralRealization	@param sessionContext @param collateralRealizationDTO @return @throws Exception	
IAccountingAuditAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingAuditAdapter	addAccountingAuditEntry		
IAccountingAuditAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingAuditAdapter	addAccountingAuditEntry		This method will log the changed fields into Audit Item Log table which will be used to maintain the   history of the individual fields changed during subsequent maintenances.  Only the changed fields will be logged, for create operation only not null values will be logged.  @param fieldComparisonResult @param domainObjectDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param auditItem @throws FatalException
IStatementInquiryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IStatementInquiryAdapter	inquireDDTransactions		
IStatementInquiryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IStatementInquiryAdapter	inquireTDTTransactions		
IStatementInquiryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IStatementInquiryAdapter	inquireLoanTransactions		
IAccountingSuspenseReconJournalEntryAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\accounting\IAccountingSuspenseReconJournalEntryAdapter	createJournalEntries	This method calls the application service of Journal MultiEntry module, and passes the newly created Journal Multi-Entry batch for further processing. It is here that the actual <b> Inter-Module Communication </b> takes place, as by means of the adapter, from Accounting Module, we are making a call to the Journal Entry module. @param sessionContext @param suspenseInquiryDTO @throws FatalException	This represents the addManualSuspenseRecon method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext,This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,SuspenseReconciliationDTO[],suspenseInquiryArrayDTO TODO: Explain the SuspenseReconciliationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IRecipientDerivationAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ep\IRecipientDerivationAdapter	fetchApprovalGroupMembers	Fetch Approval Groups from SMS @param approverGroup @return @throws FatalException	Fetch Approval Groups from SMS @param approverGroup @return @throws FatalException
ISettlementBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementBropAdapter	fetchDenominationsForGivenCurrency	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException	
ISettlementBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementBropAdapter	fetchDenominationsForGivenCurrency	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException	
ISettlementBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementBropAdapter	fetchDenominationsForGivenCurrency	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException	
ISettlementBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementBropAdapter	fetchDenominationsForGivenCurrency	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException
ISettlementBropAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementBropAdapter	fetchDenominationsForGivenCurrency	This method is used to fetch denomination details for the given currency code. @param currencyCode @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITitleSearchPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\lcm\collaterals\titlesearch\ITitleSearchPartyAdapter	fetchPartyAddress		Fetch Party Address @param sessionContext @param partyAddressDTO @return partyAddressResponse @throws FatalException
IDelayedTransactionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IDelayedTransactionAdapter	approveWorkItem	This operation is used for approving a WorkItem. @param sessionContext @param workflowId @throws FatalException	This method will process all the WorkItems, which are generated for delayed transactions. First it will fetch all such relevant workflow ids and then pass it to adapter for further approval and processing of the transaction.
IDelayedTransactionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IDelayedTransactionAdapter	approveWorkItem	This operation is used for approving a WorkItem. @param sessionContext @param workflowId @throws FatalException	This method will approve the workflow. @param sessionContext @param workflowId
ITaxationInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationInterestAdapter	processTaxRefund	This method acts as a bridge between Taxation module and Interest module This method will update the Tax refunded amount for Particular tax event in the log item. In case the Tax amount deducted for a particular event is refunded to the Customer , Then Tax module will intimate the Interest module to update the Tax-Refund amount. This updation is required because in Interest Certificate generation the Interest amount and the Tax amount is fetched from the Log Item .	This method is used to refund the tax amount to the customer. This method will take the {link AccountTaxRefundDetailsDRO} as input. User has multiple options to choose to refund the amount. @param taxRefundDTO @param sessionContext

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ITaxationInterestAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\taxation\ITaxationInterestAdapter	processTaxRefund	This method acts as a bridge between Taxation module and Interest module This method will update the Tax refunded amount for Particular tax event in the log item.  In case the Tax amount deducted for a particular event is refunded to the Customer , Then Tax module will intimate the Interest module to update the Tax-Refund amount.  This updation is required because in Interest Certif-icate generation the Interest amount and the Tax amount is fetched from the Log Item .	This method acts as a bridge between Taxation module and Interest module  {@link InterestOperationService#processTaxRefund}. 
IAssetClassificationSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAssetClassificationSecurityAdapter	fetchAssetClassificationReason		Method to get the Asset Classification Code For Party.
IAssetClassificationSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAssetClassificationSecurityAdapter	fetchAssetClassificationReason		Method to get the Asset Classification Code Description For Party.
IAssetClassificationSecurityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IAssetClassificationSecurityAdapter	fetchAssetClassificationReason		Method to get the Asset Classification Reason For Party.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentAccountAdapter	fetchAccountLevelPreferences	This method call the account module to get account level preferences for document delivery @param sessionContext This parameter captures the basic information about the user of the system. @param partyId Identifier of Party for which account information needs to be found out. @param accountId Identifier of account for which information needs to be found out. @param documentTypeId identifier of document type @return DocumentDeliveryPreferencesDTO @throws com.ofss.fc.infra.exception.FatalException	Gets Delivery preferences i.e. preferred delivery channel and destination address based   delivery channel. 1. call content to account adapter to get account level preference   2. if account level preference are not set call content to party adapter to get   party level preferences.  3. get default delivery channel defined at document type level and default party   destination corresponding to the channel   @param sessionContext   @param documentDetailDTO   @return DocumentDeliveryPreferencesResponse   @throws FatalException
IDocumentAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentAccountAdapter	fetchDeliveryPreferences	Fetch delivery preferences. @param sessionContext the session context @param documentTypeRelationshipDTOSet the document type relationship dto set @param sourcePartyId the source party id @param accountId the account id @param accountType the account type @return the list< document delivery preference dt o> @throws FatalException the fatal exception	Fetch delivery preferences. @param sessionContext the session context @param documentTypeRelationshipDTOSet the document type relationship dto set @param accountId the account id @param accountType the account type @param sourcePartyId the source party id @return the list< document delivery preference dt o> @throws FatalException the fatal exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\content\IDocumentAccountAdapter	fetchDeliveryPreferences	Fetch delivery preferences. @param sessionContext the session context @param documentTypeRelationshipDTOSet the document type relationship dto set @param sourcePartyId the source party id @param accountId the account id @param accountType the account type @return the list< document delivery preference dt o> @throws FatalException the fatal exception	Gets Delivery preferences i.e. preferred delivery channel and destination address based   delivery channel. 1. call content to account adapter to get account level preference   2. if account level preference are not set call content to party adapter to get   party level preferences.  3. get default delivery channel defined at document type level and default party   destination corresponding to the channel   @param sessionContext   @param documentDetailDTO   @return DocumentDeliveryPreferencesResponse   @throws FatalException
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	invokeNewAutoRefinancePolicy		This method is used to create a new Automatic Refinancing policy for a product group. It will make a call to the third party interface that actually creates the Auto refinance policy. Since the response is an asynchronous callback, an update service will be provided to the third party. The initial status of the policy is set to 'Pending' in origination. The policy version no. would ideally be 1. Thereafter on successful policy creation, the Auto Refinance details and the status are saved in origination. If the policy creation is unsuccessful for some reason, the corresponding status and the reason for failure are also stored in Origination. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param RefinanceProductGroupLinkageDTO refinanceProductGroupLinkageDTO, The OriginationAutoRefiInterface will need some other info like the primary applicant's financial details, the loan amount etc to construct the requestDTO to be passed to the third party interface which'll be provided by RefinanceProductGroupLinkageDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@return Transaction status @exception OriginationErrorConstants.OR_AUTO_REFI_NOT_APPLICABLE : FC_OR_AR_0001   This exception occurs when Refinance Policy is initiated for a Product Group  for which Fast Refi flag is unchecked.
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	invokeChangeRequestForAutoRefiPolicy		This method is used to create a new Automatic Refinancing policy for a product group. It will make a call to the third party interface that actually creates the Auto refinance policy. Since the response is an asynchronous callback, an update service will be provided to the third party. The initial status of the policy is set to 'Pending' in origination. The policy version no. would ideally be 1. Thereafter on successful policy creation, the Auto Refinance details and the status are saved in origination. If the policy creation is unsuccessful for some reason, the corresponding status and the reason for failure are also stored in Origination. @param SessionContext sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param RefinanceProductGroupLinkageDTO refinanceProductGroupLinkageDTO, The OriginationAutoRefiInterface will need some other info like the primary applicant's financial details, the loan amount etc to construct the requestDTO to be passed to the third party interface which'll be provided by RefinanceProductGroupLinkageDTO @return Transaction status @exception OriginationErrorConstants.OR_AUTO_REFI_NOT_APPLICABLE : FC_OR_AR_0001   This exception occurs when Refinance Policy is initiated for a Product Group  for which Fast Refi flag is unchecked.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	invokeAmendVariationRequestForAutoRefiPolicy		This method is used to create a new Automatic Refinancing policy for a product group in case of Amend. It will make a call to the third party interface that actually creates the Auto refinance policy. Since the response is an asynchronous callback, an update service will be provided to the third party. The initial status of the policy is set to 'Pending' in origination. Thereafter on successful policy creation, the Auto Refinance details and the status are saved in origination. If the policy creation is unsuccessful for some reason, the corresponding status and the reason for failure are also stored in Origination. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param RefinanceProductGroupLinkageDTO refinanceProductGroupLinkageDTO, The OriginationAutoRefiInterface will need some other info like the primary applicant's financial details, the loan amount etc to construct the requestDTO to be passed to the third party interface which'll be provided by RefinanceProductGroupLinkageDTO @return Transaction status
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	updateAutoRefinancePolicyDetails		The auto refi policy creation process is asynchronous. This method is called by the third party once the policy is successfully created. It updates the policy details in origination. If the policy creation is unsuccessful, the reason code and the status is stored in Origination Auto Refi Details Data entity @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param AutoRefinanceDetailsDTO autoRefinanceDetailsDTO, This contains all the details like OFI Payout figure, policy provider, policy status, OFI account no, etc.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	updateAutoRefinancePolicyDetails		<p>This method is called after the Generation &amp; Dispatch of documents to inform the Automatic Refinance third party interface that all the documents have been dispatched to the customer. The third party interface updates the document status for the auto refi policy in its system. Once all the LRM policies within an application has been rejected/approved, the user will generate and dispatch all the related documents together. Using the application Id, we send the Document Sent Status to third party for the policies linked it to. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, This number denotes the refinance policy number @return Transaction status @exception OriginationErrorConstants.OR_AUTO_REFI_POLICY_NOT_FOUND : FC_OR_AR_0003 &lt;br&gt; This exception occurs when trying to send DOC_SENT status to third party &lt;br&gt; for an application which does not have any refinance policy.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	updateDocSentStatusForAutoRefiPolicy		<p>This method is called after the Generation &amp; Dispatch of documents to inform the Automatic Refinance third party interface that all the documents have been dispatched to the customer. The third party interface updates the document status for the auto refi policy in its system. Once all the LRM policies within an application has been rejected/approved, the user will generate and dispatch all the related documents together. Using the application Id, we send the Document Sent Status to third party for the policies linked it to. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param String applicationId, This number denotes the refinance policy number @return Transaction status @exception OriginationErrorConstants.OR_AUTO_REFI_POLICY_NOT_FOUND : FC_OR_AR_0003 &lt;br&gt; This exception occurs when trying to send DOC_SENT status to third party &lt;br&gt; for an application which does not have any refinance policy.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IOriginationAutoRefinanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\origination\IOriginationAutoRefinanceAdapter	cancelAutoRefinancePolicy		<p>This method is used to cancel the initiated 'automatic refinance' policy creation process It will make a call to the third party interface that actually cancels the Auto refinance policy. Since the response is an asynchronous callback,the status of the policy is set to 'Cancellation Pending' in origination. After the completion of cancellation process by the third party,the asynchronous call back response is sent to OriginationAutoRefi Interface. The Status in origination will be updated to Cancelled.The cancelled status will enable the User to go with normal create offer process. @param SessionContext, sessionContext Session Context which contains information such as user id, branch, branch code, channel etc. @param AutoRefinanceDetailsDTO autoRefinanceDetailsDTO, This contains all the details like OFI Payout figure, policy provider, policy status, OFI account no, etc. @return Transaction status @exception</p> <p>OriginationErrorConstants.OR_AUTO_REFI_CANCEL_NOT_APPLICABLE : FC_OR_AR_0002 &lt;br&gt; This exception occurs when trying to cancel an already SETTLED Refinance Policy. @exception</p> <p>OriginationErrorConstants.OR_AUTO_REFI_POLICY_NOT_FOUND : FC_OR_AR_0003 &lt;br&gt; This exception occurs when trying to cancel a refinance policy for a facility &lt;br&gt; for which policy has not been initiated.</p>
ILoanDDAAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAAAdapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	@param applicationContext @param changeOverDraftToTermLoanRequestDTO @return ChangeOverDraftToTermLoanResponse
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	<pre>This method is used to fetch the Transaction Accounts . &lt;br&gt; The following steps are carried out in this method:- &lt;br&gt; 1) Call is given to ILoanPartyInformationApplicationAdapter.fetchAllAccountsOfPartySet to fetch all the transaction account number of Loan's party holder/s 2) Call is given to ILoanDDAAadapter.inquireDDAAccount to fetch the details of the Transaction Accounts. </pre> @param sessionContext @param loanAccountId @throws FatalException @returns BasicInquiryDetailsDTO[]
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchAccountBalances	The method will fetch the balances for the given casa account @param accountId @return	This method is used to fetch the balance details for given account @param sessionContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	inquireDDAAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	inquireDDAAccount		This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	inquireDDAAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	inquireDDAAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	inquireDDAAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	inquireDDAAccount		<pre>This method is used to fetch the Transaction Accounts . &lt;br&gt; The following steps are carried out in this method:- &lt;br&gt; 1) Call is given to ILoanPartyInformationApplicationAdapter.fetchAllAccountsOfPartySet to fetch all the transaction account number of Loan's party holder/s 2) Call is given to ILoanDDAA dapter.inquireDDAAccount to fetch the details of the Transaction Accounts. </pre> @param sessionContext @param loanAccountId @throws FatalException @returns BasicInquiryDetailsDTO[]

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchIsODAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchIsODAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	deLinkODandDeriveBalances	This method is called to de-link Overdraft facility from CASA account and derive balances. @param accountId @param pricefeatureDTO @return	@param applicationContext @param changeOverDraftToTermLoanRequestDTO @return ChangeOverDraftToTermLoanResponse
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	deLinkODandDeriveBalances	This method is called to de-link Overdraft facility from CASA account and derive balances. @param accountId @param pricefeatureDTO @return	@see ILoanDDAA dapter#deLinkODandDeriveBalances(String, WrappedPriceFeatureDTO)
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	addAccountFacilityLinkage		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	addAccountFacilityLinkage		<pre>This method is used to attach OD facility to transaction Account. &lt;br&gt; The following steps are carried out in this method:- &lt;br&gt; 1) AccountFacilityLinkageDTO is prepared by calling method prepareAccountFacilityLinkageDTO&lt;br&gt; 2) Call is given to addAccountFacilityLinkage method of AccountFacilityLinkageApplicationService class via LoanDDAA dapter.&lt;br&gt; </pre> @param sessionContext @param LendingVariationRequestDTO @throws



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
ILoanDDAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanDDAAadapt er	fetchBalanc eDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanDDAAadapt er	fetchBalanc eDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanDDAAadapt er	fetchBalanc eDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \loan\ILoanDDAAadapt er	fetchBalanc eDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchBalanceDetails		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchBalanceDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchBalanceDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of installment payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchBalanceDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of advance payment by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanInstallmentPaymentApplicationService that does the transaction of partial payoff by either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method takes the simplified DTO loanTransactionRequestDTO and based on the minimum user inputs for amount, mode of payment etc., invokes the appropriate API inside LoanRedrawApplicationService that does the transaction of loan redraw to either a DDA account, cash or external account. It also fetches the operating instruction and transaction fee details to be displayed on screen.
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method redeems a deposit in a Term Deposit account, the deposit can be redeemed to 1) CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method creates a new deposit in a Term Deposit account funded 1) By CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input. @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method redeems a deposit in a Term Deposit account, the deposit can be redeemed to 1) CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchBalanceDetails		This method creates a new deposit in a Term Deposit account funded 1) By CASH 2) Single Demand Deposit Account 3) External Account This is a wrapper method which internally calls the base method. It in addition expects the Account NickName as input. @param sessionContext @param TDPayinRequestDTO @return FundNewTDResponse @throws FatalException
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAAadapter	fetchEconomicCostQuote		This method inquires basic details required to process change overdraft to term loan. 1.Inquire the MOF product group and retain the list of allowed product groups. 2.Inquire the individual product groups and check if allowed product class is LN and retain the list. Fetch the od limit of the CASA account set it as loan amount. @param applicationContext @param offerId @param casaAccountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	fetchEconomicCostQuote		This method returns economic cost quote and fixed rate details. @param applicationContext @param accountId @return
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	validateDDAStatus		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	validateDDAStatus		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	getAccrualStatusForODAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	getAccrualStatusForODAccount		
ILoanDDAA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\loan\ILoanDDAA dapter	getAccrualStatusForODAccount		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingPollerAdapter	initiatePricingPoller	This method will initiate pricing poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	This method will initiate pricing poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException @see com.ofss.fc.app.adapter.pricing.IPricingPollerAdapter #initiatePricingPoller(com.ofss.fc.app.context.SessionContext)
IPricingPollerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingPollerAdapter	initiatePricingPoller	This method will initiate pricing poller. @param sessionContext the session context @throws com.ofss.fc.infra.exception.FatalException	Start all asynchronous pollers supported in the product. Currently supports pricing poller, event poller, delayed transaction poller, FTP webservice poller. Please add more in this method Additionally we also have an extension for consulting pollers @param sessionContext
IDSLAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to validate account details. @param businessContext specifies party and account details. @param partyAccountDetailsChangedDTO specifies PartyAccountDetailsChangedDTO. @param identifier specifies identifier. @param accountId specifies account id. @param accountAdapter specifies IDSLAccountAdapter. @return @throws FatalException
IDSLAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to fetch NamedPartyRelationshipDTO using identifier. @param identifier @return @throws FatalException
IDSLAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to fetch party details using nickname

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to fetch details for Loan account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to fetch details for TD account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	The following method is used to fetch details for DDA account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	@param accountId @return @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	<p>/ (non-Javadoc)</p> <p>@seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService #</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)</p>	The following method is used to validate account details. @param businessContext specifies party and account details. @param partyAccountDetailsChangeDTO specifies PartyAccountDetailsChangeDTO. @param identifier specifies identifier. @param accountId specifies account id. @param accountAdapter specifies IDSLAccountAdapter. @return @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	<p>/ (non-Javadoc)</p> <p>@seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService #</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)</p>	The following method is used to fetch NamedPartyRelationshipDTO using identifier. @param identifier @return @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	<p>/ (non-Javadoc)</p> <p>@seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService #</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.Account</p>	The following method is used to fetch party details using nickname



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			DTO, com.ofss.fc.enumeration.party.AccountType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to fetch details for Loan account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO,	The following method is used to fetch details for TD account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			com.ofss.fc.enumeration.party.AccountType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to fetch details for DDA account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	/ (non-Javadoc) @seecom.ofss.fc.app.account.service.IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			untType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.AccountType) </pre>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccount	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.AccountType) </pre>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.ofss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.AccountType) </pre>	The following method is used to validate account details. @param businessContext specifies party and account details. @param partyAccountDetailsChangeDTO specifies PartyAccountDetailsChangeDTO. @param identifier specifies identifier. @param accountId specifies account id. @param accountAdapter specifies IDSLAccountAdapter. @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.o fss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.Acco untType) </pre>	The following method is used to fetch details for Loan account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.o fss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.Acco untType) </pre>	The following method is used to fetch details for TD account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	<pre> / (non-Javadoc) @seecom.ofss.fc.app.account.service. IAccountInquiryApplicationService # fetchAccountByAccountType(com.o fss.fc.app.context .SessionContext, com.ofss.fc.app.account.dto.Account DTO, com.ofss.fc.enumeration.party.Acco untType) </pre>	The following method is used to fetch details for DDA account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	<p>/ (non-Javadoc)</p> <p>@see com.ofss.fc.app.account.service.IAccountInquiryApplicationService</p> <p>#</p> <p>fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountUsingAccountId		<p>The following method is used to set party and account related information. @param businessContext @param partyAccountDetailsChangeDTO @param accountId @param namedPartyRelationshipDTOArray @param namedPartyRelationshipDTOForIdentifier @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	inquireDDTransactions	<p>This method provides financial and non financial transaction details within specified period on demand deposit account.&lt;br&gt; On financial or non financial transaction event, application stores transaction details in terms of {@link StatementItem}. The list of the financial or non financial events on which details are logged is fixed.&lt;/br&gt; Financial transaction generally updates {@link PrincipalBalances} of the account for example Cash withdrawal ,Cash deposit, Interest settlement etc.Non financial transaction does not update balances but they update non financial attributes of the account such {@link DemandDepositLifecycleStatus} , account address. Transaction inquiry request may come from branch banking. In case of branch banking operation executive may want to see transaction performed on particular account within specified period. . {@link TransactionInquiryDTO} contains start and end date .&lt;br&gt; &lt;br&gt;This method lists all the transactions where {@link TransactionDates#postingDate} according to filters applied . Please go through {@link StatementInquiryFilterInputDTO} for more details. &lt;br&gt; &lt;br&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Financial transaction details contains following details. &lt;ol&gt;</p> <ul style="list-style-type: none"> <li data-bbox="825 365 1287 428">&lt;li&gt;Transaction Amounts: Amount in different currencies.</li> <li data-bbox="825 428 1287 524">&lt;li&gt;Transaction key/Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction.</li> <li data-bbox="825 654 1287 782">&lt;li&gt;Service Charge Details: Pricing reference number or pricing ticket from which details can be asked from pricing module.</li> <li data-bbox="825 782 1287 846">&lt;li&gt;Transaction source details-channel and application user.</li> <li data-bbox="825 846 1287 941">&lt;li&gt;Transaction dates: Different dates indicating realization of transaction. &lt;li&gt;Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic.</li> </ul> <p>Aggregate functions have been applied on financial transactions like Total amount credited, Total amount debited in given period.</p> <p>&lt;/ol&gt; Non financial transaction contain following details &lt;ol&gt;</p> <ul style="list-style-type: none"> <li data-bbox="825 1201 1287 1265">&lt;li&gt;Event date : Date on which event is performed. &lt;li&gt;Event description : description of the event like "Credit Interest rate is changed from 5% to 4.5%" &lt;/ol&gt; &lt;p&gt; &lt;br&gt;</li> </ul> <p>Following chart shows example of sample transaction inquiry response.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Input details are Account :            DDA_1,Period : '1-Jan-2010' to '15-Jan-2010' &lt;table border=1 &gt; &lt;tr&gt;            &lt;td&gt;&lt;b&gt; Account No &lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt; Posting Date&lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt;Transaction &lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt;Credit/Debit&lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt;Transaction channel&lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt;Cheque Number &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt;Transaction Amount&lt;/b&gt;&lt;/td&gt;            &lt;td&gt;&lt;b&gt;Current Balance&lt;/b&gt;&lt;/td&gt;            &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt;            &lt;td&gt;1-Jan-2010&lt;/td&gt; &lt;td&gt;Cash Deposit&lt;/td&gt; &lt;td&gt;Cr&lt;/td&gt;            &lt;td&gt;Branch&lt;/td&gt; &lt;td&gt;&lt;/td&gt;            &lt;td&gt;1000 INR&lt;/td&gt; &lt;td&gt;5000 INR&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;            &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jan- 2010&lt;/td&gt; &lt;td&gt;Cash Withdrawal&lt;/td&gt; &lt;td&gt;Dr&lt;/td&gt;            &lt;td&gt;Branch&lt;/td&gt; &lt;td&gt;&lt;/td&gt;            &lt;td&gt;2000 INR&lt;/td&gt; &lt;td&gt;3000 INR&lt;/td&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt;            &lt;td&gt;15-Jan-2010&lt;/td&gt; &lt;td&gt;Cheque deposit&lt;/td&gt; &lt;td&gt;Cr&lt;/td&gt;            &lt;td&gt;Branch&lt;/td&gt;            &lt;td&gt;101010&lt;/td&gt; &lt;td&gt;5000 INR&lt;/td&gt; &lt;td&gt;8000 INR&lt;/td&gt;            &lt;tr&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; Total credit : 6000 INR&lt;br&gt; Total Debit : 2000 INR &lt;br&gt; &lt;br&gt; Non financial event can be displayed in following manner for same inputs. &lt;table</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> border=1&gt; &lt;tr&gt; &lt;td&gt;&lt;b&gt; Account No &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt; Event Date &lt;/b&gt;&lt;/td&gt; &lt;td&gt;&lt;b&gt; Event description &lt;/b&gt;&lt;/td&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jun- 2010&lt;/td&gt; &lt;td&gt;Credit Interest rate is changed from 5% to 7%&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;DDA_1&lt;/td&gt; &lt;td&gt;10-Jun-2010&lt;/td&gt; &lt;td&gt;Account branch is changed from Vile Parle to Andheri.&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;br&gt; &lt;br&gt; @param transactionInquiryDTO MI,{@link TransactionInquiryDTO},transaction InquiryDTO, It contains the following attributes: &lt;ol&gt; &lt;li&gt;accountId: String MI it is the id of the account for which DD transactions are inquired for.&lt;/li&gt; &lt;li&gt; statementPeriodType: {@link StatementPeriodType} MI denotes the period type for which the inquiry is done&lt;/li&gt; &lt;li&gt; statementInquiryFilterInputDTO: {@link StatementInquiryFilterInputDTO}, it contains the following attributes: &lt;ol&gt; &lt;li&gt;statementInquiryFilterTypes: {@link StatementInquiryFilterType}[] it has all the filters that need to be applied.Optional&lt;/li&gt; &lt;li&gt;startAmount: {@link </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>MoneyDTO) MI if amount range filter is chosen. Denotes the start amount range.&lt;/li&gt;  &lt;li&gt;endAmount: {@link MoneyDTO) MI if amount range filter is chosen. Denotes the end amount range.&lt;/li&gt;  &lt;li&gt;checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.&lt;/li&gt; &lt;li&gt; checkNumberEnd: String MI if cheque range filter is chosen. Denotes the end range.&lt;/li&gt;  &lt;li&gt;transactionDescription: String MI if transaction description filter is chosen. Denotes the description for which the inquiry is to be made&lt;/li&gt; &lt;li&gt; transactionMnemonics: String MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;lastNumberOfTransactions: int. Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs.  &lt;li&gt; lastNumberOfDays: int. This denotes the last number of days for which the transactions are to be fetched. If this is set to zero, then</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>default values will be picked up from statement parameters. &lt;/ol&gt;  &lt;/li&gt; &lt;li&gt;startDate :{@link Date} Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods. Also if either of Last_X_Transaction or Last_N_Days filter are used, it can be null. It is a mandatory input for all other cases. &lt;/li&gt; &lt;li&gt; endDate : {@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.&lt;/li&gt; &lt;li&gt;isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.&lt;/li&gt;  &lt;/ol&gt; @return {@link DDTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link DDStatementItemDTO}.Each DDStatementItemDTO can be of following type. &lt;ol&gt; &lt;li&gt;{@link DDFinancialStatementItemDTO}: Demand deposit financial Item.- Contains financial details. &lt;li&gt;</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>{@link DDNonFinanciaStatementItemDTO} : Demand deposit non financial Item.- Contains non financial details. &lt;li&gt; {@link DDConsolidatedFinancialStatementItemDTO}: Consolidated {@link DDFinancialStatementItemDTO}. Please refer {@link ConsolidationType}. &lt;/ol&gt; All the different types of DDStatementItemDTO are combined and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.&lt;/br&gt;&lt;/br&gt; The following sections indicate the completion of the transactions with reasons: &lt;/br&gt;&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS} &lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR} &lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR} &lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_FILTER_CAN_NOT_BE_USED_ERR</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> OR} &lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ START_DATE_GREATER_THAN_ END_DATE} &lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ START_AMOUNT_GREATE_THA N_END_AMOUNT} &lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_ CHEQUE_NUMBER_START_GREA TE_THAN_CHEQUE_NUMBER_E ND} &lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ FILTER_MISMATCH_ERROR}&lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ END_DATE_AFTER_PROCESS_DA TE_ERROR} &lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ DATE_RANGE_PERIOD_TOO_LO NG_ERROR} &lt;/li&gt; &lt;li&gt;  {@link AccountErrorConstants#AC_STMT_ START_DATE_TOO_OLD_ERROR} &lt;/li&gt; &lt;/ol&gt; @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	inquireTDT ransactions	<p>This method provides financial and non financial transaction details for input filter provided on term deposit account. On financial or non financial transaction event, application stores transaction details in terms of <a href="#">{@link StatementItem}</a>.</p> <p>The list of the financial or non financial events on which details are logged is fixed. Financial transaction generally updates balances of the account for example TD payin, redemption, Interest settlement etc. Non financial transaction does not update balances but they update non financial attributes of the account account address. Transaction inquiry request may come from branch banking or direct banking.</p> <p><b>&lt;br&gt;</b> <b>&lt;ol&gt;</b> <b>&lt;li&gt;</b>In case of branch banking operation executive may want to see transaction performed on particular account within specified period.<b>&lt;/li&gt;</b> <b>&lt;li&gt;</b>In case of direct banking ,customer may want to see transactions performed on account. <a href="#">{@link TransactionInquiryDTO}</a> contains start and end date .This method lists all the transactions where <a href="#">{@link TransactionAmount#transactionDate}</a> is greater or equal than <a href="#">{@link TransactionInquiryDTO#startDate}</a> and <a href="#">{@link</a></p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>TransactionAmount#transactionDate) is less than or equal to {@link TransactionInquiryDTO#endDate} . Please refer {@link StatementInquiryFilterInputDTO} for more details .&lt;/li&gt; &lt;/ol&gt; &lt;br&gt; Financial transaction details contains following details. &lt;ol&gt; &lt;li&gt;Transaction Amounts: Amount in different currencies.&lt;/li&gt; &lt;li&gt;Transaction key/Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction.&lt;/li&gt; &lt;li&gt;Service Charge Details: Pricing reference number or pricing ticket from which details can be asked from pricing module.&lt;/li&gt; &lt;li&gt;Transaction source details-channel and application user.&lt;/li&gt; &lt;li&gt;Transaction dates: Different dates indicating realization of transaction.&lt;/li&gt; &lt;li&gt;Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic. &lt;br&gt; &lt;/ol&gt; &lt;br&gt; Non financial transaction contain following details &lt;ol&gt; &lt;li&gt;Event date : Date on which event is performed.&lt;/li&gt; &lt;li&gt;Event description : description of the event like "Interest rate is changed from</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>5% to 4.5%"&lt;/li&gt; &lt;/ol&gt; @param transactionInquiryDTO MI,{@link TransactionInquiryDTO},transaction InquiryDTO, It contains the following attributes: &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;accountId: String MI it is the id of the account for which TD transactions are inquired for.&lt;/li&gt;</li> <li>&lt;li&gt;statementPeriodType: {@link StatementPeriodType} MI denotes the period type for which the inquiry is done&lt;/li&gt;</li> <li>&lt;li&gt;statementInquiryFilterInputDT O: {@link StatementInquiryFilterInputDTO}, it contains the following attributes: &lt;ol&gt;</li> <li>&lt;li&gt;statementInquiryFilterTypes: {@link StatementInquiryFilterType}[] it has all the filters that need to be applied.Optional&lt;/li&gt;</li> <li>&lt;li&gt;startAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the start amount range. &lt;/li&gt;</li> <li>&lt;li&gt;endAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the end amount range.&lt;/li&gt;</li> <li>&lt;li&gt;checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.&lt;/li&gt;</li> <li>&lt;li&gt;checkNumberEnd: String MI if cheque range filter is chosen.</li> </ul>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Denotes the end range.&lt;/li&gt;  &lt;li&gt;transactionDescription: String  MI if transaction description filter is chosen. Denotes the description for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;transactionMnemonics: String  MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;lastNumberOfTransactions: int.  Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs.  &lt;li&gt;lastNumberOfDays: int. This denotes the last number of days for which the transactions are to be fetched. If this is set to zero, then default values will be picked up from statement parameters. &lt;/ol&gt;  &lt;/li&gt; &lt;li&gt;startDate :{@link Date}  Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods. Also if either of Last_X_Transaction or Last_N_Days filter are used, it can</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>be null. It is a mandatory input for all other cases.</p> <ul style="list-style-type: none"> <li>endDate : {@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.</li> <li>isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.</li> </ul> <p>@return {@link TDTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link TDStatementItemDTO}.Each TDStatementItemDTO can be of following type.</p> <ul style="list-style-type: none"> <li>{@link TDFinancialStatementItemDTO}: Term deposit financial Item.- Contains financial details.</li> <li>{@link TDNonFinanciaStatementItemDTO}: Term deposit non financial Item.- Contains non financial details.</li> <li>{@link TDConsolidatedFinancialStatementItemDTO}: Consolidated {@link TDFinancialStatementItemDTO}. Please refer {@link ConsolidationType}.</li> </ul> <p>All the different types of TDStatementItemDTO are combined</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.</p> <p>The following sections indicate the completion of the transactions with reasons:</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</a></li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li><a href="#">{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</a></li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li><a href="#">{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_FILTER_CAN_NOT_BE_USED_ERROR}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_START_AMOUNT_GREATER_THAN_END_AMOUNT}</a></li> <li><a href="#">{@link AccountErrorConstants#AC_STMT_CHEQUE_NUMBER_START_GREATER_THAN_CHEQUE_NUMBER_END}</a></li> </ol>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountErrorConstants#AC_STMT_FILTER_MISMATCH_ERROR}&lt;/li&gt;  &lt;li&gt; {@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	inquireLoanTransactions	<p>This method provides financial and non financial transaction details for input filter provided on term deposit account. On financial or non financial transaction event, application stores transaction details in terms of {@link StatementItem}.&lt;/br&gt; The list of the financial or non financial events on which details are logged is fixed. Financial transaction generally updates balances of the account for example TD payin, redemption, Interest settlement etc. Non financial transaction does not update balances but they update non financial attributes of the account account address. Transaction inquiry request may come from branch banking or direct banking.&lt;br&gt; &lt;ol&gt; &lt;li&gt;In case of branch</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>banking operation executive may want to see transaction performed on particular account within specified period.</p> <ul style="list-style-type: none"> <li>In case of direct banking ,customer may want to see transactions performed on account. <a href="#">{@link TransactionInquiryDTO}</a> contains start and end date .This method lists all the transactions where <a href="#">{@link TransactionAmount#transactionDate}</a> is greater or equal than <a href="#">{@link TransactionInquiryDTO#startDate}</a> and <a href="#">{@link TransactionAmount#transactionDate}</a> is less than or equal to <a href="#">{@link TransactionInquiryDTO#endDate}</a> . Please refer <a href="#">{@link StatementInquiryFilterInputDTO}</a> for more details .</li> </ul> <p>Financial transaction details contains following details. </p> <ul style="list-style-type: none"> <li>Transaction Amounts: Amount in different currencies.</li> <li>Transaction key/Audit Details: Unique information to identify transaction (transactionReferenceNumber+ subsequence number).Using this number we can inquire further details about transaction.</li> <li>Service Charge Details: Pricing reference number or pricing ticket from which details can be asked from pricing module.</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>&lt;li&gt;Transaction source details-channel and application user.&lt;/li&gt;  &lt;li&gt;Transaction dates: Different dates indicating realization of transaction.&lt;/li&gt; &lt;li&gt;Transaction Event Details: Event such as credit fund, In terms of transaction mnemonic. &lt;br&gt; &lt;/ol&gt; &lt;br&gt; Non financial transaction contain following details &lt;ol&gt; &lt;li&gt;Event date : Date on which event is performed.&lt;/li&gt; &lt;li&gt;Event description : description of the event like "Interest rate is changed from 5% to 4.5%"&lt;/li&gt; &lt;/ol&gt; @param transactionInquiryDTO MI,{@link TransactionInquiryDTO},transactionInquiryDTO, It contains the following attributes: &lt;ol&gt;  &lt;li&gt;accountId: String MI it is the id of the account for which TD transactions are inquired for.&lt;/li&gt;  &lt;li&gt;statementPeriodType: {@link StatementPeriodType} MI denotes the period type for which the inquiry is done&lt;/li&gt;  &lt;li&gt;statementInquiryFilterInputDTO: {@link StatementInquiryFilterInputDTO}, it contains the following attributes: &lt;ol&gt;  &lt;li&gt;statementInquiryFilterTypes: {@link StatementInquiryFilterType}[] it has all the filters that need to be</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>applied.Optional&lt;/li&gt;  &lt;li&gt;startAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the start amount range. &lt;/li&gt;  &lt;li&gt;endAmount: {@link MoneyDTO} MI if amount range filter is chosen. Denotes the end amount range.&lt;/li&gt;  &lt;li&gt;checkNumberStart: String MI if cheque range filter is chosen. Denotes the start range.&lt;/li&gt;  &lt;li&gt;checkNumberEnd: String MI if cheque range filter is chosen. Denotes the end range.&lt;/li&gt;  &lt;li&gt;transactionDescription: String MI if transaction description filter is chosen. Denotes the description for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;transactionMnemonics: String MI if transaction mnemonics filter is chosen. Denotes the mnemonics for which the inquiry is to be made&lt;/li&gt;  &lt;li&gt;lastNumberOfTransactions: int. Denotes last number of transaction to be inquired for in case of last x transactions filter. If this value is set to zero, the default values will be picked up from statement parameters. This is mostly used for mini statement inquiry in ATMs.  &lt;li&gt;lastNumberOfDays: int. This denotes the last number of days for</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>which the transactions are to be fetched. If this is set to zero, then default values will be picked up from statement parameters. &lt;/ol&gt;</p> <p>&lt;/li&gt; &lt;li&gt;startDate :{@link Date} Denotes the value date range start parameter. It need not be passed if the StatementPeriodType is either current or prior as in this cases the period is got from {@link StatementRequestApplicationService} methods. Also if either of Last_X_Transaction or Last_N_Days filter are used, it can be null. It is a mandatory input for all other cases.&lt;/li&gt; &lt;li&gt;endDate :{@link Date} Denotes the value date range end parameter. It is mandatory for the same cases as the start date.&lt;/li&gt; &lt;li&gt;isAscending : boolean MI denotes if the inquired statement is to be sorted in ascending or descending order.&lt;/li&gt; &lt;/ol&gt; @return {@link LoanTransactionInquiryResponse}. The response contains the requested statement items for the given filters. It also contains the required statistics{@link StatementStatistics}.Response contains array of {@link LoanStatementItemDTO}.Each LoanStatementItemDTO can be of following type. &lt;ol&gt; &lt;li&gt;{@link LoanFinancialStatementItemDTO}:</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Loan financial Item.- Contains financial details. &lt;/ol&gt; All the different types of LoanStatementItemDTO are combined and given in the array, sorted on the value dates in case of Financial statements and the event date in case of non financial statements.&lt;/br&gt;&lt;/br&gt; The following sections indicate the completion of the transactions with reasons: &lt;/br&gt;&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt; {@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_FILTER_CAN_NOT_BE_USED_ERROR}&lt;/li&gt; . &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_GREATER_THAN_END_DATE}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_AMOUNT_GREATER_THAN_END_AMOUNT}&lt;/li&gt; &lt;li&gt; {@link</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>AccountErrorConstants#AC_STMT_CHEQUE_NUMBER_START_GREATER_THAN_CHEQUE_NUMBER_END}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_FILTER_MISMATCH_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_END_DATE_AFTER_PROCESS_DATE_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_DATE_RANGE_PERIOD_TOO_LONG_ERROR}&lt;/li&gt; &lt;li&gt; {@link AccountErrorConstants#AC_STMT_START_DATE_TOO_OLD_ERROR}&lt;/li&gt; &lt;/ol&gt; @throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	searchDDTransactions	<p>This method returns the transaction status for all the DD financial items. This method provides the feature of oracle text for fetching the information @param SessionContext sessionContext @param String accountId @param String queryPhrase @param PageInformationDTO pageInformation @return DDTransactionInquiryResponse @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	searchTDTransactions	This method returns the transaction status for all the TD financial items. This method provides the feature of oracle text for fetching the information @param SessionContext sessionContext @param String accountId @param String queryPhrase @param PageInformationDTO pageInformation @return DDTransactionInquiryResponse @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	searchLoan Transactions	This method returns the transaction status for all the Loan financial items. This method provides the feature of oracle text for fetching the information @param SessionContext sessionContext @param String accountId @param String queryPhrase @param PageInformationDTO pageInformation @return DDTransactionInquiryResponse @throws FatalException	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	/(non-Javadoc) @see com.ofss.fc.app.account.service.IAccountInquiryApplicationService#fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to validate account details. @param businessContext specifies party and account details. @param partyAccountDetailsChangeDTO specifies PartyAccountDetailsChangeDTO. @param identifier specifies identifier. @param accountId specifies account id. @param accountAdapter specifies IDSLAccountAdapter. @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	/(non-Javadoc) @see com.ofss.fc.app.account.service.IAccountInquiryApplicationService#fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to fetch details for Loan account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	/(non-Javadoc) @see com.ofss.fc.app.account.service.IAccountInquiryApplicationService#fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to fetch details for TD account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	/(non-Javadoc) @see com.ofss.fc.app.account.service.IAccountInquiryApplicationService#fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	The following method is used to fetch details for DDA account. @param contextResponse @param businessContext @param stringBuilder @param dslAccountAdapter @throws FatalException
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountByAccountType	/(non-Javadoc) @see com.ofss.fc.app.account.service.IAccountInquiryApplicationService#fetchAccountByAccountType(com.ofss.fc.app.context.SessionContext, com.ofss.fc.app.account.dto.AccountDTO, com.ofss.fc.enumeration.party.AccountType)	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			untType)	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountAddressLinkage	<p>This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAINquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If</p>	<p>This method is used to issue adhoc statement of the customer account by calling the respective CASA service. @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for adhoc request @return TransactionStatus if adhoc request logging to CASA failed or was success</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>the account status is closed or dormant , " Closed/Dormant Account " error message would be displayed. 6) Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>partyId as input to get the relationship. d) For each docTypeId in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountAddressLinkage	<p>This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAINquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant</p>	<p>This method is used to issue adhoc statement of the customer account by calling the respective TD service. @param sessionContext the session context to be used in interacting with the service @param accountId the account id to be used for adhoc request @return TransactionStatus if adhoc request logging to CASA failed or was success</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Account " error message would be displayed. 6) Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountAddressLinkage	<p>This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAINquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant</p>	<p>This method is used to issue adhoc statement of the customer account by calling the respective Loan service. @param sessionContext the session context to be used in interacting with the service @param accountID the account id to be used for adhoc request @return TransactionStatus if adhoc request logging to CASA failed or was success</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Account " error message would be displayed. 6) Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchAccountAddressLinkage	<p>This method fetches the address linkages for all the parties associated with the account. In the Inquiry DTO the partyId and docTypeId fields would not contain any value. Steps to be performed are 1) Create an instance of AccountAddressLinkageInquiryResponse. 2) Fetch the transaction status using fetchTransactionStatus(). 3) Create Transaction Context by passing SessionContext and INQUIRE (as MaintenanceType) as an arguments. 4) Based on the account type , check the validity of the account and get its status by calling the appropriate Application Service through the adapter. If the account is not valid , "Invalid Account Number" error message is displayed. If the account type is CASA , call fetchBasicDetails() method exposed by the DDAInquiryApplicationService with the accountId as input. If the account type is TD , call fetchAccountDetails() method exposed by the TermDepositAccountApplicationService with the accountId as input. 5) Call validateAccount() method with accountId and status as the input. If the account status is closed or dormant , " Closed/Dormant</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Account " error message would be displayed. 6) Create an instance of the Domain Service (AccountAddressLinkageService). 7) Call the fetchAccountAddressLinkage() method of the AccountAddressLinkageService with accountId as the input. 8) Receive the AccountAddressLinkage entity in response. 9) Based on this response, call different Application Services to fetch other relevant details through the adapter. a) For each party in the AccountAddressLinkage, call the fetchPartyDetails() of PartyApplicationService with partyId as the input to get party details. b) For each party in the AccountAddressLinkage, i) For all the postal address preference set in the linkage call the fetchPartyAddress() of PartyAddressApplicationService to with partyId and postalAddressType to get party postal address. c) For each party in the Account Address Linkage, call the fetchPartyAccountRelationship() of PartyAccountRelationshipApplicationService with accountId and partyId as input to get the relationship. d) For each docTypeId</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>in the linkage, call the fetchDocumentType() of DocumentTypeApplicationService with docTypeId as the input to get the description of the document. 10) Fill the AccountAddressLinkageInquiryResponse with all the relevant details. 11) Convert the AccountAddressLinkageInquiryResponse entity to DTO using the Assembler. 12) Return the AccountAddressLinkageInquiryResponseDTO. @param accountAddressLinkageInquiryDTO The partyId field would be null @param sessionContext</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderDDStatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these</p>	<p>The following method is used to generate DDA Statement. @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link BulkStatementRequest}. In this case service charge for {@link BulkStatementRequest} will be for all individual requests. Generated statement will be delivered using external delivery system, While raising request {@link DeliverySuccessDetailsDTO} and {@link DeliveryFailureDetailsDTO} should be blank. Statement request id is system generated so no need to provide it. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>ddStatementRequestDTO  MI, {@link  DDStatementRequestDTO} It  contains all the information required  to place a demand deposit  statement request. &lt;/br&gt;Its  attributes are: &lt;ol&gt; &lt;li&gt;key :{@link  StatementRequestKey} contains the  statement id. It needs to be null as  the statement id will be generated  by the service.&lt;/li&gt; &lt;li&gt;accountId :  String MI, it denotes the demand  deposit account id for which the  statement request is to be  made.&lt;/li&gt;  &lt;li&gt;statementPeriodType : {@link  StatementPeriodType} MI. Denotes  the period type of the input.&lt;/li&gt;  &lt;li&gt;period :{@link Period} MI if  statementPeriodType is equal to  TRANSACTION. It contains the  start and the end date for statement  request.&lt;/li&gt;  &lt;li&gt;statementGenerationDate :  {@link Date} must be null.&lt;/li&gt;  &lt;li&gt;pricing :{@link  ServiceChargeDetails} contains the  service charge details for the  statement request.&lt;/li&gt;  &lt;li&gt;statementRequestType :{@link  StatementRequestType},MI, Denotes  the request type.&lt;/li&gt;  &lt;li&gt;statementRequestStatus :{@link  StatementRequestStatus} must be  null&lt;/li&gt; &lt;li&gt;statementNo : Must</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>be set to null for statement order.</p> <ul style="list-style-type: none"> <li>customerInitiatedRequest: {@link CustomerInitiatedRequest} MI if statementRequestType is IndividualStatementRequest or BulkStatmentRequest. It contains all the customer initiated request details.</li> </ul> <p>@return {@link TransactionStatus}</p> <p><b>Transaction Success</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> <p><b>Transaction Failure</b></p> <ul style="list-style-type: none"> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> <p>Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li>{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}</li> </ol> <p>@throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderDDStatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link BulkStatementRequest}. In this case service charge for {@link BulkStatementRequest} will be for all individual requests. Generated statement will be delivered using external delivery system, While raising request {@link DeliverySuccessDetailsDTO} and {@link DeliveryFailureDetailsDTO} should be blank. Statement request id is system generated so no need to</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>provide it. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param ddStatementRequestDTO MI,{@link DDStatementRequestDTO} It contains all the information required to place a demand deposit statement request. &lt;br&gt;Its attributes are: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt; &lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;statementGenerationDate : {@link Date} must be null.&lt;/li&gt; &lt;li&gt;pricing :{@link ServiceChargeDetails} contains the service charge details for the</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre> statement request.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Denotes the request type.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest : {@link CustomerInitiatedRequest} MI if statementRequestType is IndividualStatementRequest or BulkStatmentRequest. It contains all the customer initiated request details.&lt;/li&gt; &lt;/ol&gt; @return {@link TransactionStatus} &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#FATAL_INT_ERROR }&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ ACC_BLANK_INPUT_ERROR}&lt;/li &gt; &lt;/ol&gt; &lt;/ul&gt; @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderTDStatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order term deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link BulkStatementRequest}. In</p>	The following method is used to generate TD Statement. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>this case service charge for {@link BulkStatementRequest} will be for all individual requests. Generated statement will be delivered using external delivery system, While raising request {@link DeliverySuccessDetailsDTO} and {@link DeliveryFailureDetailsDTO} should be blank. Statement request id is system generated so no need to provide it. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param ddStatementRequestDTO MI, {@link DDStatementRequestDTO} It contains all the information required to place a demand deposit statement request. &lt;/br&gt;Its attributes are: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt;</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>&lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt;</p> <p>&lt;li&gt;statementGenerationDate : {@link Date} must be null.&lt;/li&gt;</p> <p>&lt;li&gt;pricing :{@link ServiceChargeDetails} contains the service charge details for the statement request.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Denotes the request type.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null&lt;/li&gt;</p> <p>&lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;</p> <p>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI if statementRequestType is IndividualStatementRequest or BulkStatmentRequest. It contains all the customer initiated request details.&lt;/li&gt;</p> <p>&lt;/ol&gt;</p> <p>@return {@link TransactionStatus} &lt;/br&gt;</p> <p>&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt;</p> <p>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;</p> <p>&lt;/ul&gt;</p> <p>&lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt;</p> <p>&lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following</p> <ol style="list-style-type: none"> <li>&lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt;</li> </ol> <p>&gt; &lt;/ol&gt; &lt;/ul&gt; @throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderTDSstatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order term deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request. In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link BulkStatementRequest}. In this case service charge for {@link BulkStatementRequest} will be for all individual requests. Generated statement will be delivered using external delivery system, While raising request {@link DeliverySuccessDetailsDTO} and {@link DeliveryFailureDetailsDTO} should be blank. Statement request id is system generated so no need to provide it. @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param ddStatementRequestDTO MI, {@link DDStatementRequestDTO} It contains all the information required to place a demand deposit statement request. &lt;/br&gt;Its attributes are: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>by the service.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>&lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>&lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.</li> <li>&lt;li&gt;statementGenerationDate : {@link Date} must be null.</li> <li>&lt;li&gt;pricing :{@link ServiceChargeDetails} contains the service charge details for the statement request.</li> <li>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Denotes the request type.</li> <li>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null</li> <li>&lt;li&gt;statementNo : Must be set to null for statement order.</li> <li>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI if statementRequestType is IndividualStatementRequest or BulkStatmentRequest. It contains all the customer initiated request details.</li> </ul> <p>@return {@link TransactionStatus}</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt;  &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;  &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt; &lt;/ol&gt; &lt;/ul&gt; @throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderLoanStatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise</p>	<p>The following method is used to generate Loan Statement. @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in <a href="#">{@link CustomerInitiatedRequest#bulkRequests}</a>. In this case service charge for <a href="#">{@link CustomerInitiatedRequest#bulkRequests}</a> will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. <code>@fcb.param MI,SessionContext,sessionContext, ,</code> This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. <code>@fcb.param MI, <a href="#">{@link LoanStatementRequestDTO}</a>,loanStatementRequestDTO , ,</code> It contains all the information required to place a demand deposit statement request.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>For a customer initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.</li> <li>accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.</li> <li>partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.</li> <li>statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.</li> <li>period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.</li> <li>actionDueDate : {@link Date} must be null. Will be the current process day.</li> <li>actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.</li> <li>pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>this service.&lt;/li&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt;</li> <li>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt;</li> <li>&lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;</li> <li>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.&lt;/li&gt;</li> </ul> <p>&lt;/ol&gt;</p> <p>&lt;/br&gt;For a System initiated request, the attributes need to be as follows:</p> <ul style="list-style-type: none"> <li>&lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt;</li> <li>&lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt;</li> <li>&lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt;</li> <li>&lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt;</li> </ul>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>&lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt; &lt;li&gt;actionDateDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.&lt;/li&gt; &lt;li&gt;actionDatePickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt; &lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.&lt;/li&gt; &lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending&lt;/li&gt; &lt;li&gt;statementNo : Must be set to null for statement order.It will be set at statement generation.&lt;/li&gt; &lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} Must be null&lt;/li&gt; &lt;/ol&gt; @return {@link LoanStatementRequestInquiryResponse}. It contains the {@link LoanStatementRequestDTO} of the</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>demand deposit statement request that was ordered. It also contains the transaction status of this transaction. The following sections indicate the completion of the transactions with reasons &lt;/br&gt;</p> <p>&lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt;  &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;  &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; &lt;ul&gt;  Error Reason may be one of the following &lt;ol&gt; &lt;li&gt;{@link AccountErrorConstants#AC_STMT_ACC_BLANK_INPUT_ERROR}&lt;/li&gt;  &gt; &lt;/ol&gt; @throws FatalException</p>	
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	orderLoanStatement	<p>System using statement set up done at product/offering level can initiate statement request System Initiated Request. Customer can raise ad-hoc statement request {@link CustomerInitiatedRequest}. Customer can reorder existing generated/requested statement. This method is used to order demand deposit account statement. System periodically raise statement request according statement setup done. Statement setup contains statement generation frequency. Although statement setup contains delivery</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>mode, statement period, service charge details etc, we store all these details in the statement request as statement setup may modify in future. Customer can raise statement request .In this case, delivery mode, statement period will be input from customer. Operations manager can modify/waive service charges given by pricing module. Customer can reorder statement. In this case customer may choose different delivery mode and address. Service charge in this case may differ from original statement request. Customer can also raise multiple statement requests at same time. System will store this request in {@link CustomerInitiatedRequest#bulkRequests}. In this case service charge for {@link CustomerInitiatedRequest#bulkRequests} will be for all individual requests. Generated statement will be delivered using external delivery system. Statement request id is system generated so no need to provide it. @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>purposes. @fcb.param MI, {@link LoanStatementRequestDTO},loanStatementRequestDTO , , It contains all the information required to place a demand deposit statement request.</p> <p>&lt;/br&gt;For a customer initiated request, the attributes need to be as follows: &lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the statement request is delivery is to be done.&lt;/li&gt;</p> <p>&lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt;</p> <p>&lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION and not bulk reorder. It contains the start and the end date for statement request.&lt;/li&gt;</p> <p>&lt;li&gt;actionDate : {@link Date} must be null. Will be the current process day.&lt;/li&gt;</p> <p>&lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>working day.&lt;/li&gt; &lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be either IndividualStatementRequest or BulkStatmentRequest for customer intimated request.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} MI. In case of Statement delivery mode as hand delivery, this will be executed. Other wise it must be null. If null it will be set to pending.&lt;/li&gt;</p> <p>&lt;li&gt;statementNo : Must be set to null for statement order.&lt;/li&gt;</p> <p>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest} MI. It contains all the customer initiated request details.&lt;/li&gt; &lt;/ol&gt;</p> <p>&lt;/br&gt;For a System initiated request, the attributes need to be as follows:</p> <p>&lt;ol&gt; &lt;li&gt;key :{@link StatementRequestKey} contains the statement id. It needs to be null as the statement id will be generated by the service.&lt;/li&gt; &lt;li&gt;accountId : String MI, it denotes the demand deposit account id for which the statement request is to be made.&lt;/li&gt; &lt;li&gt;partyId : String MI, it denotes the party id for which the</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>statement request is delivery is to be done.&lt;/li&gt;</p> <p>&lt;li&gt;statementPeriodType : {@link StatementPeriodType} MI. Denotes the period type of the input.&lt;/li&gt;</p> <p>&lt;li&gt;period :{@link Period} MI if statementPeriodType is equal to TRANSACTION. It contains the start and the end date for statement request.&lt;/li&gt;</p> <p>&lt;li&gt;actionDateDate : {@link Date} MI denotes the date on which the statement generation action is to be logged.&lt;/li&gt;</p> <p>&lt;li&gt;actionPickupDate : {@link Date} must be null. Will be same as the actionPickupDate if actionPickupDate is not on a holiday. Else will be the next working day.&lt;/li&gt;</p> <p>&lt;li&gt;pricing :{@link FeeDetailsDTO} contains the service charge details for the statement request. Only the pricingTicketNumber is tracked by this service.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestType :{@link StatementRequestType},MI, Must be set to SystemInitiatedRequest.&lt;/li&gt;</p> <p>&lt;li&gt;statementRequestStatus :{@link StatementRequestStatus} must be null. It will be set to pending&lt;/li&gt;</p> <p>&lt;li&gt;statementNo : Must be set to null for statement order.It will be set at statement generation.&lt;/li&gt;.</p> <p>&lt;li&gt;customerInitiatedRequest :{@link CustomerInitiatedRequest}</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Must be null&lt;/li&gt; &lt;/ol&gt; @return  {@link  LoanStatementRequestInquiryResponse}. It contains the {@link  LoanStatementRequestDTO} of the  demand deposit statement request  that was ordered. It also contains the  transaction status of this  transaction. The following sections  indicate the completion of the  transactions with reasons &lt;/br&gt;  &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt;  &lt;li&gt;{@link  com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}&lt;/li&gt;  &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction  Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link  com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}&lt;/li&gt;  &lt;/ul&gt; &lt;/ul&gt; &lt;ul&gt;  Error Reason may be one of the  following &lt;ol&gt; &lt;li&gt;{@link  AccountErrorConstants#AC_STMT_&lt;br&gt; ACC_BLANK_INPUT_ERROR}&lt;/li&gt;  &gt; &lt;/ol&gt; @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchCurrentStatementPeriod	<p>Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. &lt;/br&gt;Consider the following example.</p> <pre>&lt;/br&gt;AccountId: TestAccount001 &lt;/br&gt;Opening date: 22nd-feb-2008 &lt;/br&gt;Statements generated: &lt;table border='1'&gt; &lt;tr&gt; &lt;th&gt;Statement no.&lt;/th&gt; &lt;th&gt;Statement period start&lt;/th&gt; &lt;th&gt;Statement period end&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;nth&lt;/td&gt; &lt;td&gt;1-april-2008&lt;/td&gt; &lt;td&gt;30-april-2008&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1&lt;/td&gt; &lt;td&gt;22-feb-2008&lt;/td&gt; &lt;td&gt;29-feb-2008&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; Suppose the current process date is 24th may 2008.&lt;/br&gt;&lt;/br&gt;Then the current period: 1-may-2008 to 24th-may-2008 &lt;/br&gt;Prior statement period is: 1-april-2008 to 30-april-2008 &lt;/br&gt;&lt;/br&gt;In case there are no</pre>	<p>The following method returns the appended text that needs to be used in case of oracle text based search</p> <pre>@param sessionContext @param transactionInquiryDTO @param accountType @param accountDTO @return @throws FatalException</pre>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>statements generated till 24th may 2008, then &lt;/br&gt;Current statement period: 22nd-feb-2008(opening date) to 24th may 2008 &lt;/br&gt;Prior statement period:22nd-feb-2008(opening date) to 24th may 2008&lt;p&gt;This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as follows: &lt;ol&gt; &lt;li&gt;start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not be after the current process date. &lt;li&gt;end date: Its the current process date. &lt;/ol&gt;</p> <p>@param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param accountId MI,String, It is the account id of the account for which statement period is to be fetched. @param accountType MI,{@link com.ofss.fc.enumeration.account.AccountType}, It denotes the type of</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit. @return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons</p> <pre> &lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#FATAL_INT_ERROR }&lt;/li&gt; &lt;/ul&gt; @throws FatalException </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDSLAccount Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dsl\IDSLAccountAdapter	fetchCurrentStatementPeriod	<p>Statement inquiry and statement request have the facility of fetching items that are in the current statement period or prior statement period. Current period is the duration between the latest endDate of a system initiated statement request that has been generated and the current process date. Prior period is the same period as the last system initiated statement request that was generated. &lt;/br&gt;Consider the following example.</p> <p>&lt;/br&gt;AccountId: TestAccount001          &lt;/br&gt;Opening date: 22nd-feb-2008          &lt;/br&gt;Statements generated: &lt;table border='1'&gt; &lt;tr&gt; &lt;th&gt;Statement no.&lt;/th&gt; &lt;th&gt;Statement period start&lt;/th&gt; &lt;th&gt;Statement period end&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1&lt;/td&gt; &lt;td&gt;22-feb-2008&lt;/td&gt; &lt;td&gt;29-feb-2008&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;2&lt;/td&gt; &lt;td&gt;1-april-2008&lt;/td&gt; &lt;td&gt;30-april-2008&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;td&gt;...&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;1&lt;/td&gt; &lt;td&gt;22-feb-2008&lt;/td&gt; &lt;td&gt;29-feb-2008&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; Suppose the current process date is 24th may 2008.&lt;/br&gt;&lt;/br&gt;Then the current period: 1-may-2008 to 24th-may-2008 &lt;/br&gt;Prior statement period is: 1-april-2008 to 30-april-2008 &lt;/br&gt;&lt;/br&gt;In case there are no</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>statements generated till 24th may 2008, then &lt;/br&gt;Current statement period: 22nd-feb-2008(opening date) to 24th may 2008 &lt;/br&gt;Prior statement period:22nd-feb-2008(opening date) to 24th may 2008&lt;p&gt;This method will fetch the current statement period for the given account id and account type. The current statement period can be derived as follows: &lt;ol&gt; &lt;li&gt;start date: It is the day after the latest endDate of all the statement request that have been generated for the particular account which are system initiated. If there has been no such statement requests in the account, the value considered will be from the account opening date. start date value can not be after the current process date. &lt;li&gt;end date: Its the current process date. &lt;/ol&gt;</p> <p>@param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param accountId MI,String, It is the account id of the account for which statement period is to be fetched. @param accountType MI,{@link com.ofss.fc.enumeration.account.AccountType}, It denotes the type of</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>account that is required. It can be DemandDeposit or ExistingTermDeposit or NewTermDeposit. @return {@link StatementPeriodInquiryResponse}. It contains the required current statement period. It also contains the transaction status. The following sections indicate the completion of the transactions with reasons</p> <pre>&lt;/br&gt; &lt;b&gt;Transaction Success&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#SUCCESS}&lt;/li&gt; &lt;/ul&gt; &lt;/br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt; &lt;ul&gt; &lt;li&gt;{@link com.ofss.fc.infra.enumeration.Respo nseCodeType#FATAL_INT_ERROR }&lt;/li&gt; &lt;/ul&gt; @throws FatalException</pre>	
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isValidOfferCombinatorForEvent	<p>This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed.</p> <p>@param fromOfferId @param toOfferId @param eventId @return</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isValidOfferCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromOfferId @param toOfferId @param eventId @return	
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isValidOfferCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromOfferId @param toOfferId @param eventId @return	
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isValidOfferCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromOfferId @param toOfferId @param eventId @return	This method is used to do inter product fund transfer check. @param fromOfferId @param toOfferId @param eventCode

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isValidOfferCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromOfferId @param toOfferId @param eventId @return	
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isFetchProductCodeForAccount	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @return	This method is used to fetch the product code using the account id. @param accountId @param settlementModeType @return @throws FatalException
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isFetchProductCodeForAccount	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @return	This method is used to fetch the product code using the account id. @param accountId @param settlementModeType @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ISettlementProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\settlement\ISettlementProductAdapter	isFetchProductCodeForAccount	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @return	
IAccountingCASABalanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingCASABalanceAdapter	fetchCASAPrevDayBalance		@param appContext @param untankAdjData @throws FatalException @throws RuntimeException @throws CloneNotSupportedException
IAccountingCASABalanceAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\accounting\IAccountingCASABalanceAdapter	fetchCASACurrentBalance	This method is used to fetch the current balance of the Demand Deposit account. The balance obtained is the accounting balance of the Demand Deposit account. @param accountIdArray @return Map<String,CASAAmountTagDetails> @since 2.3	This method is used to check and create adjustment entries for the reversal transaction to the Demand deposit account. This is required as the balances might be in asset or liability and also may be in Normal or Suspended state. The method goes through the entries that were created earlier and in case of demand deposit account forms the ReversalAdjustmentDataDetails and checks whether adjustment entries are required to ensure that the balances are in sync. @param acctEvent @throws CloneNotSupportedException @throws RuntimeException @throws FatalException
ITDSMSAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\td\ITDSMSAdapter	fetchAllNonFinancialTaskCodes	Method to return non-financial task codes @param sessionContext @return List<String> @throws FatalException	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
ICollectionPartyDemographicsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\collection\ICollectionPartyDemographicsAdapter	updateIndividualDemographics	This method is used to modify individual party demographics details. @param partyDemographicsDTO @return	This method is used to modify individual party demographics details. @param partyDemographicsDTO @return
IDDAOpsAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAOpsAdapter	isBodRunForToday		This will check whether bod has run for today in this environment.If the batch mode is on then it won't check because batch mode is on means batch is already running.This requires in case of online transaction which doesn't know batch is running or not. @return
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	@param sessionContext @param absentTellerActivityDTOs @return @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param absentTellerActivityDTO @param taskCode @param requesterId @param requestedId @param transactionRefNo @param actionInitiatedBy @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchDTO the <tt>CurrencyMismatchDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	Once an indent is been raised all the updation and changes or maintaining approval or rejection of currency indent is been done by this method. This method will update any changes made on a raised indent in currency indent table and changed record is been added to currency indent history table for auditing. If the indent is been updated from NextGen screen before any action on it is been taken by the approver in workflow, the old process is been terminated and a new process is been initiated. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CurrencyIndentDTO,currencyIndentDTO, ,currencyIndentDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyIndentDTO @param taskCode @param isRequester @param sessionContext @param branchID @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the exchange method of the entity. The method will be used for denomination exchange in the same currency. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CashPositionDTO,transactionUnits, ,transactionUnits ,,The object contains the transaction units during denomination exchange @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the transfer method of the entity. The method will perform the task of debiting/crediting provided ledger account and crediting/debiting CASH GL. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,MiscLedgerTransferDetailsDTO,miscLedgerTransferDetailsDTO, ,This parameter captures the basic information about transaction. This information will be utilized to transfer cash between requesting and requested users. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the addCurrencyMismatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CurrencyMismatchDTO,currencyMismatchDTO, ,currencyMismatchDTO @return {@link com.ofss.fc.app.brop.dto.currencyMismatch.CurrencyMismatchBookingResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the reverseCurrencyMismatch method of the entity. @param sessionContext the <tt>SessionContext</tt> object. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object having the reversal details. @return transaction status object <tt>TransactionStatus</tt>. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the reverseCurrencyMismatch method of the entity. @param sessionContext the <tt>SessionContext</tt> object. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object having the reversal details. @return transaction status object <tt>TransactionStatus</tt>. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchDTO the <tt>CurrencyMismatchDTO</tt> object. @param taskCode the task code/fast path. @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the openBranchBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the openTellerBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the closeBranchBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the closeTellerBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This represents the transfer method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BranchCurrencyProcessorDTO,branchCurrencyProcessorDTO, ,This parameter captures the basic information about transaction. This information will be utilized to transfer cash between requesting and requested users. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>BranchCurrencyProcessorDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	fetchTellerTransactionLog	This method will fetch the <tt>DebitCreditDetailsDTO</tt> object for a given search criteria. @param searchDTO the <tt>TransactionLogSearchDTO</tt> DTO containing the search criteria. @return the <tt>DebitCreditDetailsDTO</tt> object.	This represents the fetchDenominationAndTransactionDetails method of the entity. The method will fetch cash position and transaction details (number of debits, number of credits and the amounts for the same) for a given branch, user, posting date and currency code. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.   The information is used by the system for authentication purposes. @fcb.param MI,com.ofss.fc.app.brop.dto.cashposition.CashPositionDTO,cashPositionDTO, ,cashPositionDTO @fcb.param MI,com.ofss.fc.enumeration.brop.UserType,userType, ,userType @return {@link com.ofss.fc.app.brop.dto.cashposition.CashTransactionInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	fetchTellerTransactionLog	This method will fetch the <tt>DebitCreditDetailsDTO</tt> object for a given search criteria. @param searchDTO the <tt>TransactionLogSearchDTO</tt> DTO containing the search criteria. @return the <tt>DebitCreditDetailsDTO</tt>	This method is used to return the cash position for the current posting date. Additionally, it will return the cash position upper and lower bounds which are user specific. @param sessionContext @param transactionLogSearchDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			object.	
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	@param sessionContext @param absentTellerActivityDTOs @return @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param absentTellerActivityDTO @param taskCode @param requesterId @param requestedId @param transactionRefNo @param actionInitiatedBy @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchDTO the <tt>CurrencyMismatchDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	Once an indent is been raised all the updation and changes or maintaining approval or rejection of currency indent is been done by this method. This method will update any changes made on a raised indent in currency indent table and changed record is been added to currency indent history table for auditing. If the indent is been updated from NextGen screen before any action on it is been taken by the approver in workflow, the old process is been terminated and a new process is been initiated. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CurrencyIndentDTO,currencyIndentDTO, ,currencyIndentDTO @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyIndentDTO @param taskCode @param isRequester @param sessionContext @param branchID @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the exchange method of the entity. The method will be used for denomination exchange in the same currency. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CashPositionDTO,transactionUnits, ,transactionUnits ,,The object contains the transaction units during denomination exchange @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the transfer method of the entity. The method will perform the task of debiting/crediting provided ledger account and crediting/debiting CASH GL. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,MiscLedgerTransferDetailsDTO,miscLedgerTransferDetailsDTO, ,This parameter captures the basic information about transaction. This information will be utilized to transfer cash between requesting and requested users. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the addCurrencyMismatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,CurrencyMismatchDTO,currencyMismatchDTO, ,currencyMismatchDTO @return {@link com.ofss.fc.app.brop.dto.currencyMismatch.CurrencyMismatchBookingResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the reverseCurrencyMismatch method of the entity. @param sessionContext the <tt>SessionContext</tt> object. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object having the reversal details. @return transaction status object <tt>TransactionStatus</tt>. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the reverseCurrencyMismatch method of the entity. @param sessionContext the <tt>SessionContext</tt> object. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object having the reversal details. @return transaction status object <tt>TransactionStatus</tt>. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchDTO the <tt>CurrencyMismatchDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyMismatchReversalDTO the <tt>CurrencyMismatchReversalDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the openBranchBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the openTellerBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the closeBranchBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the closeTellerBatch method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BatchOperationDTO,batchOperationDTO, ,batchOperationDTO TODO: Explain the batchOperationDTO above @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This represents the transfer method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,BranchCurrencyProcessorDTO,branchCurrencyProcessorDTO, , This parameter captures the basic information about transaction. This information will be utilized to transfer cash between requesting and requested users. @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTransactionLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>BranchCurrencyProcessorDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	fetchTellerBatchOperationDetails	This method fetches teller batch operation details for given date @param userNo @param transactionBranch @param postingDate @return @throws FatalException	@param sessionContext @param batchOperationDTO @return @throws com.ofss.fc.infra.exception.FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	fetchTellerBatchOperationDetails	This method fetches teller batch operation details for given date @param userNo @param transactionBranch @param postingDate @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) and entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) and entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog	This method is used to add (log) an entry in Teller Transaction Log Table. @param transactionLogDTO @param actionInitiatedBy @return @throws FatalException	This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>BranchCurrencyProcessorDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>MiscLedgerTransferDetailsDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param tellerTransactionLogItemDTO the <tt>TellerTransactionLogItemDTO</tt> object. @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param batchOperationDTO the <tt>BatchOperationDTO</tt> object. @param taskCode the task code. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) and entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) and entry in Teller Transaction Log Table. @param transactionLogDTO the <tt>TransactionLogDTO</tt> object
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	addTellerTransactionActivityLog		This method is used to add (log) an entry in Teller Transaction Log Table. @param currencyProcessorDTO the <tt>BranchCurrencyProcessorDTO</tt> object. @param taskCode the task code/fast path. @throws FatalException
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	executePreferenceOnline		This method logs accounting entries while booking and reversing currency mismatch through settlement. @param currencyMismatchDTO the <tt>CurrencyMismatchDTO</tt> object. @param transactionDetailsDTOs the <tt>TransactionDetailsDTO</tt> array. @param transactionReferenceNumber the transactionReferenceNumber @param u FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	executePreferenceOnline		This method will be used for the execution of the chosen settlement mode. The method will in turn call the settlement service <a href="#">{@link SettlementExecutionService#executePreferenceOnline}</a> This method is used in case the component preference does not exist. The settlement service in this case will create a component preference and then execute the preference.
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	validateDraftsStatus		
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	validateDraftsStatus		
IBranchAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\brop\IBranchAccountAdapter	validateDraftsStatus		
IAgentProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentProductAdapter	fetchLoanOfferDTO		This service will return AccountSummaryDTO for all the account under facilityId provided in input. @param facilityId :facility id of which all child Account will be returned back @return AccountSummaryResponse
IAgentProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\agent\IAgentProductAdapter	fetchCASA OfferDTO		This service will return AccountSummaryDTO for all the account under facilityId provided in input. @param facilityId :facility id of which all child Account will be returned back @return AccountSummaryResponse

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">TicketService#startDialogue(Session Context)</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">PricingTicketDTO</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">PricingTicketDTO</a> must then be encapsulated into a <a href="#">PricingContextDTO</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">FatalException</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">Interaction</a> or <a href="#">com.ofss.fc.domain.pricing.service.ticket.TicketDomainService</a> <a href="#">TicketDomainService</a> throws an exception</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.&lt;br&gt; Refactored to this module from <a href="#">DemandDepositSimulationApplicationService</a> on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;br&gt; {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the {@link TicketService#startDialogue(SessionContext) } &lt;br&gt; Creates, persists, and returns a new {@link PricingTicketDTO}, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a {@link PricingTicketDTO} before calling any other methods of an application service belonging to Pricing. This {@link PricingTicketDTO} must then be encapsulated into a {@link PricingContextDTO} and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a {@link FatalException} and abort. &lt;/p&gt; @param sessionContext the session context @throws FatalException if {@link Interaction} or {@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService} throws an</p>	<p>This service will call the {@link TicketService#startDialogue(SessionContext) } &lt;br&gt; Creates, persists, and returns a new {@link PricingTicketDTO}, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a {@link PricingTicketDTO} before calling any other methods of an application service belonging to Pricing. This {@link PricingTicketDTO} must then be encapsulated into a {@link PricingContextDTO} and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a {@link FatalException} and abort. &lt;/p&gt; @param sessionContext the session context @throws FatalException if {@link Interaction} or {@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService} throws an exception</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			exception	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">TicketService#startDialogue(SessionContext)</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">PricingTicketDTO</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">PricingTicketDTO</a> must then be encapsulated into a <a href="#">PricingContextDTO</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">FatalException</a> and abort.&lt;p&gt; @param sessionContext the session context @throws FatalException if <a href="#">Interaction</a> or <a href="#">com.ofss.fc.domain.pricing.service.ticket.TicketDomainService</a> TicketDomainService) throws an exception</p>	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. The algorithm is as follows: &lt;ul&gt; If pricing ticket is not null &lt;li&gt;Indicating that the rate is already negotiated with the pricing module and same is required to be applied.&lt;/li&gt; &lt;li&gt;Call <a href="#">ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])</a>&lt;/li&gt; &lt;li&gt;pass <a href="#">EventRateVarianceDTO[], EventPriceVarianceDTO[]</a> both as null &lt;li&gt;set other attributes in the input.&lt;/ul&gt; &lt;ul&gt; If pricing ticket is null &lt;li&gt;Indicating that there is no rate communication happened between pricing and the calling module.&lt;/li&gt; &lt;li&gt;Default rate is required to be applied.&lt;/li&gt; &lt;li&gt;Find the recommended rate by calling <a href="#">ContractApplicationService#getPriceOffering()</a>&lt;/li&gt; &lt;li&gt;Call <a href="#">ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])</a> to apply the variance and creating pricing plan&lt;/li&gt; &lt;/ul&gt; &lt;/ol&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>The method is responsible for creating backdated pricing plan in the pricing module. DDAPricingPlanAdapterDTO would include the account opening date as one of the attribute along with the process date. The algorithm is as follows: If pricing ticket is not null Indicating that the rate is already negotiated with the pricing module and same is required to be applied. Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> pass <a href="#">EventRateVarianceDTO[], EventPriceVarianceDTO[]</a> both as null set other attributes in the input. If pricing ticket is null Indicating that there is no rate communication happened between pricing and the calling module. Default rate is required to be applied. Find the recommended rate by calling <a href="#">{@link ContractApplicationService#getPriceOffering()}</a> Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> to apply the variance and creating pricing plan &lt;br&gt; &lt;b&gt;Since:&lt;/b&gt; R2.2 &lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">TicketService#startDialogue(SessionContext)</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">PricingTicketDTO</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">PricingTicketDTO</a> must then be encapsulated into a <a href="#">PricingContextDTO</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">FatalException</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">Interaction</a> or <a href="#">com.ofss.fc.domain.pricing.service.ticket.TicketDomainService</a> throws an exception</p>	<pre> / &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call <a href="#">com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)</a> Method to acquire a <a href="#">PricingTicketDTO</a>. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling <a href="#">getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO</a> &lt;/li&gt; &lt;li&gt; Instantiate an object of <a href="#">PricingEventContextDTO</a> &lt;/li&gt; &lt;li&gt; Set the <a href="#">PricingTicketDTO</a> obtained earlier in this <a href="#">PricingEventContextDTO</a> &lt;/li&gt; &lt;li&gt; Instantiate <a href="#">EventPriceApplicationService</a> &lt;/li&gt; &lt;li&gt; Call the method <a href="#">EventPriceApplicationService#getPriceOnEvent()</a> which returns <a href="#">PriceFeatureDTO</a> &lt;/li&gt; &lt;li&gt; Copy the values from <a href="#">PriceFeatureDTO</a> to an array of <a href="#">PricingDetail</a> as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;li&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt; Set the various attributes of <a href="#">PricingTicketDTO</a> </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingDetail} with the attribute values present in {<a href="#">@link PriceFeatureDTO</a>} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {<a href="#">@link DepositsFeeRequestInfo</a>} for more details. &lt;/li&gt; &lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See {<a href="#">@link PricingDetail</a>} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. The method will in turn call the service <a href="#">{@link EventPriceApplicationService#getPriceForAccountClosure}</a> @param sessionContext @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	This service will call the {@link TicketService#startDialogue(SessionContext) }   Creates, persists, and returns a new {@link PricingTicketDTO}, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.<p>This method MUST be called to acquire a {@link PricingTicketDTO} before calling any other methods of an application service belonging to Pricing. This {@link PricingTicketDTO} must then be encapsulated into a {@link PricingContextDTO} and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a {@link FatalException} and abort.</p> @param sessionContext the session context @throws FatalException if {@link Interaction} or {@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService} throws an exception	This method populates pricing context without setting the facts. @param applicationContext @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>Re-prices the regular and penalty base interest index offerings in the <a href="#">{@link PricingPlan}</a> associated with the <a href="#">{@code arrangementNumber}</a>. This involves discarding all existing interest index offerings, re-evaluating the regular and penalty interest policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the regular and penalty margin index offerings in the <a href="#">{@link PricingPlan}</a> associated with the <a href="#">{@code arrangementNumber}</a>. This involves discarding all existing margin offerings, re-evaluating the regular and penalty margin policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the price offerings in the <a href="#">{@link PricingPlan}</a> associated with the <a href="#">{@code arrangementNumber}</a>. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price offerings, and are NOT reinstated with the new price offerings. &lt;p&gt; The <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)}</a> <a href="#">TicketService.startDialogue(SessionContext)}</a> method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling this method. The <a href="#">{@link PricingTicketDTO}</a> must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>passed in, the method simply throws a <a href="#">{@link FatalException}</a> and aborts.   @paramsessionContext the session context @paramsessionContext @param dDAPricingPlanAdapterDTO @throws FatalException if the input is invalid, or there is a problem while repricing</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a>   Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.   This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.   @paramsessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ti</a></p>	<p>This method would generate the pricing contract context dto using the demand deposit account as input which would then be used in re-pricing @param idemandDepositAccount @return</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			cket.TicketDomainService TicketDomainService) throws an exception	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.i nternal.interface.jar@\co m\ofss\fc\app\adapter \dda\IDDA PricingAda pter	startDialog ue	This service will call the {@link TicketService#startDialogue(Session Context) }   Creates, persists, and returns a new {@link PricingTicketDTO}, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.<p>This method MUST be called to acquire a {@link PricingTicketDTO} before calling any other methods of an application service belonging to Pricing. This {@link PricingTicketDTO} must then be encapsulated into a {@link PricingContextDTO} and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a {@link FatalException} and abort. </p> @param sessionContext the session context @throws FatalException if {@link Interaction} or {@link com.ofss.fc.domain.pricing.service.ti cket.TicketDomainService TicketDomainService) throws an exception	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's. In case of backdated account opening for periodic fees, a PeriodicEventTrackerDTO will be returned from pricing which demand deposit needs to use and log a periodic fee action.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method is to get Differed Fees for the nominated account. @param currencyCodes @param arrangementNumber @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method will be called to create and carry forward the existing UDM to new offer while offer swap @param adapterDTO @param eventRateVarianceDTOs @param forwardUDM @return ArrangementPlanLinkageKeyDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">TicketService#startDialogue(SessionContext)</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">PricingTicketDTO</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">PricingTicketDTO</a> must then be encapsulated into a <a href="#">PricingContextDTO</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">FatalException</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">Interaction</a> or <a href="#">com.ofss.fc.domain.pricing.service.ticket.TicketDomainService</a> throws an exception</p>	<pre> / &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call <a href="#">com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)</a> Method to acquire a <a href="#">PricingTicketDTO</a>. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling <a href="#">getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO</a> &lt;/li&gt; &lt;li&gt; Instantiate an object of <a href="#">PricingEventContextDTO</a> &lt;/li&gt; &lt;li&gt; Set the <a href="#">PricingTicketDTO</a> obtained earlier in this <a href="#">PricingEventContextDTO</a> &lt;/li&gt; &lt;li&gt; Instantiate <a href="#">EventPriceApplicationService</a> &lt;/li&gt; &lt;li&gt; Call the method <a href="#">EventPriceApplicationService#getPriceOnEvent()</a> which returns <a href="#">PriceFeatureDTO</a> &lt;/li&gt; &lt;li&gt; Copy the values from <a href="#">PriceFeatureDTO</a> to an array of <a href="#">PricingDetail</a> as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;li&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt; Set the various attributes of <a href="#">PricingTicketDTO</a> </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingDetail] with the attribute values present in {<a href="#">@link PriceFeatureDTO</a>} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {<a href="#">@link DepositsFeeRequestInfo</a>} for more details. &lt;/li&gt; &lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See {<a href="#">@link PricingDetail</a>} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>On OD Account opening, origination marks some of the fees as capitalised. Capitalised means fee amount will be added in OD amount. At the time of disbursement , DDA module fetch pending capitalised fees from pricing and charge it. @param accountId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(Session Context) }</a> &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> </p> @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception	<p>Method used to fetch fees based on recommendation pricing ticket number. &lt;br&gt; When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number.Following steps are performed . &lt;ol&gt;  &lt;li&gt;Create New fee ticker number. &lt;li&gt;Create DDA pricing event context. &lt;li&gt;Call Pricing service with old recommendation and new fee ticket number. &lt;/ol&gt; @param feeRequestInfo @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">TicketService#startDialogue(SessionContext)</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">PricingTicketDTO</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">PricingTicketDTO</a> must then be encapsulated into a <a href="#">PricingContextDTO</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">FatalException</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">Interaction</a> or <a href="#">com.ofss.fc.domain.pricing.service.ticket.TicketDomainService</a> throws an exception</p>	<p>In Amend OD account workflow. User can change various attributes of the account like Offer, OD amount, OD duration, Account Preferences etc. If product behaviour allows to reprice, we create new pricing recommendation based of new attributes. We create recommendation in following condition in amend workflow. &lt;ol&gt; &lt;li&gt;Offer swap is performed. &lt;li&gt;Increase / Decrease OD amount, provided bank policy permits repricing. &lt;/ol&gt; &lt;br&gt; This method is used to create new recommendation. Following are steps involved. &lt;ol&gt; &lt;li&gt;Create new Price ticket. &lt;li&gt;Create facts based on new attributes. &lt;li&gt;Create new recommendation. &lt;/ol&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method will add a broker rebate margin for an od linkage which is linked to an existing account.</p> <p>@param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> } &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>In Manage facility, Using existing facility , New OD account is created.This method is used to create new recommendation while opening account.Following are steps involved. &lt;ol&gt; &lt;li&gt;Create new Price ticket.&lt;/li&gt;&lt;li&gt;Create facts based on new attributes.&lt;/li&gt;&lt;li&gt;Create new recommendation. &lt;/ol&gt; @param sessionContext @param pricingRequestDTO @param factMap @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	startDialogue	<p>This service will call the <a href="#">{@link TicketService#startDialogue(SessionContext)}</a> &lt;br&gt; Creates, persists, and returns a new <a href="#">{@link PricingTicketDTO}</a>, which is subsequently used to uniquely identify this dialogue/caller/client to all calls to a given application service within Pricing.&lt;p&gt;This method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling any other methods of an application service belonging to Pricing. This <a href="#">{@link PricingTicketDTO}</a> must then be encapsulated into a <a href="#">{@link PricingContextDTO}</a> and passed as a parameter to all subsequent calls to the methods of the application service within the same dialogue. If no ticket or an invalid ticket is passed in, the service will throw a <a href="#">{@link FatalException}</a> and abort.</p> <p>&lt;/p&gt; @param sessionContext the session context @throws FatalException if <a href="#">{@link Interaction}</a> or <a href="#">{@link com.ofss.fc.domain.pricing.service.ticket.TicketDomainService}</a> throws an exception</p>	<p>This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRatesForDDA	<p>This service will call the <a href="#">ContractApplicationService#getRegularInterestIndexOffering</a>&lt;br&gt;Evaluates the policy chart for regular interest indices to create a <a href="#">PricingRecommendation</a>. On its evaluation, an <a href="#">EvaluatedInterestRecommendation</a> is obtained, detailing the regular interest index offering. &lt;p&gt; The <a href="#">com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)</a> <a href="#">TicketService.startDialogue(SessionContext)</a> method MUST be called to acquire a <a href="#">PricingTicketDTO</a> before calling this method. The <a href="#">PricingTicketDTO</a> must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a <a href="#">FatalException</a> and aborts. &lt;/p&gt; @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException</p>	<p>This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.&lt;br&gt; Refactored to this module from <a href="#">DemandDepositSimulationApplicationService</a> on March 27. @fcb.param MI,SessionContext,sessionContext, SessionContext which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>&lt;br&gt; {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRatesForDDA	<p>This service will call the {@link ContractApplicationService#getRegularInterestIndexOffering}&lt;br&gt; Evaluates the policy chart for regular interest indices to create a {@link PricingRecommendation}. On its evaluation, an {@link EvaluatedInterestRecommendation} is obtained, detailing the regular interest index offering. &lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to This method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. &lt;/p&gt; @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param</p>	<p>This service will call the {@link ContractApplicationService#getRegularInterestIndexOffering}&lt;br&gt; Evaluates the policy chart for regular interest indices to create a {@link PricingRecommendation}. On its evaluation, an {@link EvaluatedInterestRecommendation} is obtained, detailing the regular interest index offering. &lt;p&gt; The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to This method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts. &lt;/p&gt; @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param originationFactsDTO @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			productGroupId @param originationFactsDTO @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchMarginsForDDA	This service will call the <a href="#">{@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)}</a>   Evaluates the policy chart for regular margin indices to create a <a href="#">{@link PricingRecommendation}</a> . On its evaluation, an <a href="#">{@link EvaluatedMarginRecommendation}</a> is obtained, detailing the regular margin index offering. <p> The <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)}</a> method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling this method. The <a href="#">{@link PricingTicketDTO}</a> must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a <a href="#">{@link FatalException}</a> and aborts. </p> @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  Refactored to this module from <a href="#">{@link DemandDepositSimulationApplicationService}</a> on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			productGroupId @param originationFactsDTO @throws FatalException	NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception   {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchMarginsForDDA	<p>This service will call the <a href="#">{@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)}</a> }</p> <p>&lt;br&gt; Evaluates the policy chart for regular margin indices to create a <a href="#">{@link PricingRecommendation}</a>. On its evaluation, an <a href="#">{@link EvaluatedMarginRecommendation}</a> is obtained, detailing the regular margin index offering. &lt;p&gt; The <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)}</a> method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling this method. The <a href="#">{@link PricingTicketDTO}</a> must then be passed as a parameter to all subsequent calls to This method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a <a href="#">{@link FatalException}</a> and aborts. &lt;/p&gt; @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException</p>	<p>This service will call the <a href="#">{@link ContractApplicationService#getRegularMarginIndexOfferingWithoutEvaluation(SessionContext, PricingContractContextDTO, Date)}</a> }</p> <p>&lt;br&gt; Evaluates the policy chart for regular margin indices to create a <a href="#">{@link PricingRecommendation}</a>. On its evaluation, an <a href="#">{@link EvaluatedMarginRecommendation}</a> is obtained, detailing the regular margin index offering. &lt;p&gt; The <a href="#">{@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)}</a> method MUST be called to acquire a <a href="#">{@link PricingTicketDTO}</a> before calling this method. The <a href="#">{@link PricingTicketDTO}</a> must then be passed as a parameter to all subsequent calls to This method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a <a href="#">{@link FatalException}</a> and aborts. &lt;/p&gt; @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchServicingFeesForDDA	This service will call {@link ContractApplicationService#getPriceOfferingWithoutEvaluation}  and {@link ContractApplicationService#getRewardOfferingWithoutEvaluation}  to stamp the fee recommendation and reward offerings against the price ticket number. @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  Refactored to this module from {@link DemandDepositSimulationApplicationService} on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				  {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchServicingFeesForDDA	This service will call {@link ContractApplicationService#getPriceOfferingWithoutEvaluation}  and {@link ContractApplicationService#getRewardOfferingWithoutEvaluation}  to stamp the fee recommendation and reward offerings against the price ticket number. @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException	This service will call {@link ContractApplicationService#getPriceOfferingWithoutEvaluation}  and {@link ContractApplicationService#getRewardOfferingWithoutEvaluation}  to stamp the fee recommendation and reward offerings against the price ticket number. @param sessionContext @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param originationFactsDTO @param accountConfigurationDTO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getNetInterestRate	This service will call the {@link ContractApplicationService#fetchRatesForRecommendation(SessionContext, PricingContractContextDTO, InputStageDetailsDTO[], Date, Date)} @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param productGroupLinkageType	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  Refactored to this module from {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>@param rateType @return @throws FatalException</p>	<p>DemandDepositSimulationApplicationService} on March 27. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception &lt;br&gt; {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getNetInterestRate	This service will call the <a href="#">ContractApplicationService#fetchRatesForRecommendation(SessionContext, PricingContractContextDTO, InputStageDetailsDTO[], Date, Date)</a> @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param productGroupLinkageType @param rateType @return @throws FatalException	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. Refactored to this module from <a href="#">DemandDepositSimulationApplicationService</a> on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getNetInterestRate	This service will call the <a href="#">{@link ContractApplicationService#fetchRatesForRecommendation(SessionContext, PricingContractContextDTO, InputStageDetailsDTO[], Date, Date)}</a> @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param productGroupLinkageType @param rateType @return @throws FatalException	This service will call the <a href="#">{@link ContractApplicationService#fetchRatesForRecommendation(SessionContext, PricingContractContextDTO, InputStageDetailsDTO[], Date, Date)}</a> @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param productCode @param productGroupId @param productGroupLinkageType @param rateType @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	addUDMtoRecommendation	Refer <a href="#">{@link ContractApplicationService#updateUDMInRecommendation(SessionContext, String, EventRateVarianceDTO)}</a> <a href="#">{@link ContractApplicationService#mergeRateVarianceToRecommendation(SessionContext, PricingContractContextDTO, EventRateVarianceDTO[])}</a> for more details about this service.   This service will add the UDM for CASA or OD against the Credit or Overdraft stage. This service is called when broker gives rebate in commission and that benefit is to be propagated to the customer. @param sessionContext @param originationFactsDTO @param effectiveDate @param	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.   Refactored to this module from <a href="#">{@link DemandDepositSimulationApplicationService}</a> on March 27. @fcb.param MI,SessionContext,sessionContext,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException	MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception   {@link DemandDepositAccountErrorConstant#INTEREST_O FFERINGS_NOT_FOUND} : FC_OR_PR_0001   This error occurs when:  Interest rate offerings cannot be fetched from Pricing module. 

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	addUDMto Recommendation	Refer <a href="#">@link ContractApplicationService#updateUDMInRecommendation(SessionContext, String, EventRateVarianceDTO)</a> <a href="#">@link ContractApplicationService#mergeRateVarianceToRecommendation(SessionContext, PricingContractContextDTO, EventRateVarianceDTO[])</a> for more details about this service. This service will add the UDM for CASA or OD against the Credit or Overdraft stage. This service is called when broker gives rebate in commission and that benefit is to be propagated to the customer. @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException	This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module. Refactored to this module from <a href="#">@link DemandDepositSimulationApplicationService</a> on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	addUDMto Recommendation	Refer <a href="#">@link ContractApplicationService#updateUDMInRecommendation(SessionContext, String, EventRateVarianceDTO)</a> <a href="#">@link ContractApplicationService#mergeRateVarianceToRecommendation(SessionContext, PricingContractContextDTO, EventRateVarianceDTO[])</a> for more details about this service. This service will add the UDM for CASA or OD against the Credit or Overdraft stage. This service is called when broker gives rebate in commission and that benefit is to be propagated to the customer. @param sessionContext @param originationFactsDTO @param effectiveDate @param pricingTicketNumber @param offerId @param productGroupId @param productGroupLinkageType @param endDate @param udmValue @throws FatalException	This method will add a broker rebate margin for an od linkage which is linked to an existing account. @param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	<p>Refere <a href="#">{@link DDAPricingAdapter#createPricingPlanAndEndDialogue}</a> @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId</p>	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. It works as follows:  &lt;li&gt;Create AdpaterFactory and get an instance of DDAPricingAdapter&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link IDDA PricingAdapter#createPricingPlanAndEndDialogue}</a>(priceTicketNumber: String, accountNumber: String) to create the pricing plan&lt;/li&gt; &lt;ol&gt; Creating PricingPlan Works as follows:: &lt;ul&gt; If pricing ticket is not null &lt;li&gt;indicating that the rate is already negotiated with the pricing module and same is required to be applied.&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue}</a>(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])&lt;/li&gt; &lt;li&gt;pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null &lt;li&gt;set other attributes in the input.&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; If pricing ticket is null &lt;li&gt;Indicating that there is no rate communication happened between pricing and the calling module.&lt;/li&gt; &lt;li&gt;Default rate is required to be applied.&lt;/li&gt; &lt;li&gt;Find the recommended rate by calling <a href="#">{@link ContractApplicationService#getPriceOffering}</a>()&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue}</a>(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[]) to apply the variance and creating pricing plan&lt;/li&gt; &lt;/ul&gt; @param priceTicketNumber @param demandDepositAccount @param postingDate @param amendConfigurationDTO @param parentAccountId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	Refere {@link DDAPricingAdapter#createPricingPlanAndEndDialogue} @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	Refere {@link DDAPricingAdapter#createPricingPlanAndEndDialogue} @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. The algorithm is as follows: &lt;ul&gt; If pricing ticket is not null &lt;li&gt;indicating that the rate is already negotiated with the pricing module and same is required to be applied.&lt;/li&gt; &lt;li&gt;Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}&lt;/li&gt; &lt;li&gt;pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null &lt;li&gt;set other attributes in the input.&lt;/li&gt;&lt;/ul&gt; &lt;ul&gt; If pricing ticket is null &lt;li&gt;Indicating that there is no rate communication happened between pricing and the calling module.&lt;/li&gt; &lt;li&gt;Default rate is required to be applied.&lt;/li&gt; &lt;li&gt;Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()}&lt;/li&gt; &lt;li&gt;Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} to apply the variance and creating pricing plan&lt;/li&gt; &lt;/ul&gt; &lt;/ol&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	Refere {@link DDAPricingAdapter#createPricingPlanAndEndDialogue} @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId	The method is responsible for creating backdated pricing plan in the pricing module. DDAPricingPlanAdapterDTO would include the account opening date as one of the attribute along with the process date. The algorithm is as follows: If pricing ticket is not null Indicating that the rate is already negotiated with the pricing module and same is required to be applied. Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null set other attributes in the input. If pricing ticket is null Indicating that there is no rate communication happened between pricing and the calling module. Default rate is required to be applied. Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()} Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} to apply the variance and creating pricing plan   <b>Since:</b> R2.2  
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	Refere {@link DDAPricingAdapter#createPricingPlanAndEndDialogue} @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId	This method will be called to create and carry forward the existing UDM to new offer while offer swap @param adapterDTO @param eventRateVarianceDTOs @param forwardUDM @return ArrangementPlanLinkageKeyDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingPlanAndEndDialogue	<p>Refere <a href="#">{@link DDAPricingAdapter#createPricingPlanAndEndDialogue}</a> @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId</p>	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. It works as follows:  &lt;li&gt;Create AdpaterFactory and get an instance of DDAPricingAdapter&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link IDDA PricingAdapter#createPricingPlanAndEndDialogue}</a>(priceTicketNumber: String, accountNumber: String) to create the pricing plan&lt;/li&gt; &lt;ol&gt; Creating PricingPlan Works as follows:: &lt;ul&gt; If pricing ticket is not null &lt;li&gt;Iindicating that the rate is already negotiated with the pricing module and same is required to be applied.&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue}</a>(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])&lt;/li&gt; &lt;li&gt;pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null &lt;li&gt;set other attributes in the input.&lt;/li&gt; &lt;/ul&gt; &lt;ul&gt; If pricing ticket is null &lt;li&gt;Indicating that there is no rate communication happened between pricing and the calling module.&lt;/li&gt; &lt;li&gt;Default rate is required to be applied.&lt;/li&gt; &lt;li&gt;Find the recommended rate by calling <a href="#">{@link ContractApplicationService#getPriceOffering()}</a>&lt;/li&gt; &lt;li&gt;Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue}</a>(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[]) to apply the variance and creating pricing plan&lt;/li&gt; &lt;/ul&gt; @param priceTicketNumber @param demandDepositAccount @param postingDate @param parentAccountId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	createPricingPlanAndEndDialogue	<p>Refere <a href="#">{@link DDAPricingAdapter#createPricingPlanAndEndDialogue}</a> @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId</p>	<p>This method would call the adapter class DDAPricingAdapter. The method is responsible for creating pricing plan in the pricing module for backdated account. It works as follows: Create AdpaterFactory and get an instance of DDAPricingAdapter Call <a href="#">{@link IDDAPricingAdapter#createPricingPlanAndEndDialogue(dDAPricingPlanAdapterDTO: DDAPricingPlanAdapterDTO)}</a> The DDAPricingPlanAdapterDTO class would have the attribute for posting date and account opening date to create the pricing plan Creating PricingPlan Works as follows:: If pricing ticket is not null indicating that the rate is already negotiated with the pricing module and same is required to be applied. Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null set other attributes in the input. If pricing ticket is null Indicating that there is no rate communication happened between pricing and the calling module. Default rate is required to be applied. Find the recommended rate by calling <a href="#">{@link ContractApplicationService#getPriceOffering()}</a> Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> to apply the variance and creating pricing plan &lt;br&gt; &lt;b&gt;Since: R2.2 &lt;br&gt;</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	create PricingPlanAndEndDialogueBackDated	Refer <a href="#">{@link DDAPricingAdapter#createPricingPlanAndEndDialogueBackDated}</a> @param priceTicketNumber @param accountNumber @param priceTicketNumber @param accountCurrency @param postingDate @param partyId	This method would call the adapter class DDAPricingAdapter. The method is responsible for creating pricing plan in the pricing module for backdated account. It works as follows: Create AdpaterFactory and get an instance of DDAPricingAdapter Call <a href="#">{@link IDDA PricingAdapter#createPricingPlanAndEndDialogue(dDAPricingPlanAdapterDTO: DDAPricingPlanAdapterDTO)}</a> The DDAPricingPlanAdapterDTO class would have the attribute for posting date and account opening date to create the pricing plan Creating PricingPlan Works as follows:: If pricing ticket is not null indicating that the rate is already negotiated with the pricing module and same is required to be applied. Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null set other attributes in the input. If pricing ticket is null Indicating that there is no rate communication happened between pricing and the calling module. Default rate is required to be applied. Find the recommended rate by calling <a href="#">{@link ContractApplicationService#getPriceOffering()}</a> Call <a href="#">{@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</a> to apply the variance and creating pricing plan   <b>Since: R2.2  

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getFeeOnEvent	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. </pre> @param factMap @param feeRequestInfo	This method derives the default fees for the event. @param eventId @param accountId @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getFeeOnEvent	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. </pre> @param factMap @param feeRequestInfo	This method is responsible for charging MinimumAverageBalance fee if average balance is not maintained in the account. Assumptions: <ul><li>BalanceChangeAction will execute before MinimumAverageBalance action</li> <li>Fee collection for account closure will be handled by account closure service</li> </ul>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getFeeOnEvent	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. </pre> @param factMap @param feeRequestInfo	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getFeeOnEvent	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. </pre> @param factMap @param feeRequestInfo	/ <pre>This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} Method to acquire a {@link PricingTicketDTO}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {@link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate an object of {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Set the {@link PricingTicketDTO} obtained earlier in this {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<pre> EventPriceApplicationService} &lt;/li&gt; &lt;/li&gt; Call the method {@link EventPriceApplicationService#getPriceOnEvent()} which returns {@link PriceFeatureDTO} &lt;/li&gt; &lt;/li&gt; Copy the values from {@link PriceFeatureDTO} to an array of {@link PricingDetail} as follows: &lt;ul&gt; &lt;/li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;/li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;/li&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt; prices.(ChargeInformationDTO)chargeInformation.nsf Treatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/li&gt; Set the various attributes of {@link PricingDetail} with the attribute values present in {@link PriceFeatureDTO} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;/li&gt; ¶param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {@link DepositsFeeRequestInfo} for more details. &lt;/li&gt; &lt;/li&gt; ¶return PricingDetail[] : The fee details for the given event. See {@link PricingDetail} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getFeeOnEvent	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. @param factMap @param feeRequestInfo </pre>	<pre> This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} Method to acquire a {@link PricingTicketDTO}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {@link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate an object of {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Set the {@link PricingTicketDTO} obtained earlier in this {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate {@link EventPriceApplicationService} &lt;/li&gt; &lt;li&gt; Call the method {@link EventPriceApplicationService#getPriceOnEvent()} which returns {@link PriceFeatureDTO} &lt;/li&gt; &lt;li&gt; Copy the values from {@link PriceFeatureDTO} to an array of {@link PricingDetail} as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;li&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt; Set the various attributes of {@link </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingDetail} with the attribute values present in {<a href="#">@link PriceFeatureDTO</a>} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {<a href="#">@link DepositsFeeRequestInfo</a>} for more details. &lt;/li&gt; &lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See {<a href="#">@link PricingDetail</a>} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getFeeOnEvent	<pre>&lt;pre&gt;This method definition is used to call the actual method implementation in {<a href="#">@link PricingAdapter</a>} to get the fees for a specified event. &lt;/pre&gt; @param factMap @param feeRequestInfo</pre>	<pre>/ &lt;pre&gt;This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;/pre&gt; &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {<a href="#">@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)</a>} Method to acquire a {<a href="#">@link PricingTicketDTO</a>}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {<a href="#">@link getBankingFactContextDTO(DepositFeeRequestInfo: BankingFactContextDTO)</a>} &lt;/li&gt; &lt;li&gt; Instantiate an object of {<a href="#">@link PricingEventContextDTO</a>} &lt;/li&gt; &lt;li&gt; Set the {<a href="#">@link PricingTicketDTO</a>} obtained earlier in this {<a href="#">@link PricingEventContextDTO</a>} &lt;/li&gt; &lt;li&gt; Instantiate {<a href="#">@link EventPriceApplicationService</a>} &lt;/li&gt; &lt;li&gt; Call the method {<a href="#">@link EventPriceApplicationService#getPriceOnEvent()</a>} which returns {<a href="#">@link PriceFeatureDTO</a>} &lt;/li&gt; &lt;/ol&gt; Copy the values from {<a href="#">@link</a></pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PriceFeatureDTO} to an array of {@link PricingDetail} as follows: &lt;ul&gt; &lt;li&gt; feeAmount : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(MoneyBagDTO)netAmount &lt;/li&gt; &lt;li&gt; feeDescription : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.] &lt;/li&gt; &lt;/ul&gt; nsfTreatment : PriceFeatureDTO.List&lt;ProcessedBasePriceDTO&gt;.prices.(ChargeInformationDTO)chargeInformation.nsfTreatment &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; Set the various attributes of {@link PricingDetail} with the attribute values present in {@link PriceFeatureDTO} by iterating through the details in it. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; &amp;#064;param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See {@link DepositsFeeRequestInfo} for more details. &lt;/li&gt; &lt;li&gt; &amp;#064;return PricingDetail[] : The fee details for the given event. See {@link PricingDetail} for more details. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param feeRequestInfo @param accountCurrencyCode @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getFeeOnEvent	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to get the fees for a specified event. </pre> @param factMap @param feeRequestInfo	This method is used to fetch and calculate total fees for Account Closure Event. @param applicationContext @param accountId @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This method will delegate call to the corresponding executor for processing of the logged action.
IDDAPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This method will be used to intimate pricing module. We are using common method to intimate pricing as fees can have additional fees. @param applicationContext @param accountId @param wrappedPriceFeatureDTOList
IDDAPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	Method to credit Reward Amount On Closure @param sessionContext @param accountId @param accountTransaction @param demandDepositAccountService @param ddaPricingAdapter
IDDAPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This is an instance variable and not a class variable (static or static final). This is required to support multi-entity wide logging.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimate Pricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This method is responsible for levying Periodic Fees on the account in case bulking flag is true.   Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.   The bulking indicator and the frequency is setup in pricing module.   Once the fee details are populated by the caller method this method internally gives a call   to levyGiven which in turn will debit the account and intimate pricing.   @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimate Pricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimate Pricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This method will be used to intimate pricing module about fees charging. This is public method will be used by DDA services. @param feesChargingRequest @param wrappedPriceFeatureDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimate Pricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	This method will intimate pricing when fee amount is recovered after system hold is released from the account. @param depositsFeeRequestInfo @param transactionReferenceNumber @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. @param feeDetails </pre>	<pre> This method is internally called by {@link DDAFeesManagementService#checkAndLevyFees()}. For this method, the fees to be deducted will be provided by the caller. These fees were obtained in an earlier call to pricing where even a pricing ticket number was generated. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt;Get the alternate account for fees posting for the account on which transaction is being posted. &lt;/li&gt; &lt;li&gt;Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id. &lt;/li&gt; &lt;li&gt;If alternate account is maintained, it will be returned else original account is returned. The fees will be posted to the account returned. &lt;/li&gt; &lt;li&gt;Debit the given fees from the account by calling {@link DemandDepositAccountService#debitFunds()} &lt;/li&gt; &lt;li&gt;The fees to be deducted will be the net fee arrived at by taking into account the base fee, additional fee and the discounts applied on them. &lt;/li&gt; &lt;li&gt;Also the details of the fee deduction will be returned by {@link DemandDepositAccountService#debitFunds()} detailing how much fee was debited, earmarked or waived off. &lt;/li&gt; &lt;li&gt;The debited fee amount, earmarked fee amount and waived fee amount are set in the {@link FeeDetails} object &lt;/li&gt; &lt;li&gt;Instantiate PricingAdapterFactory. &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapterFactory#getPricingAdapter()} to get the pricing adapter instance of type {@link IPricingAdapter} &lt;/li&gt; &lt;li&gt;Call {@link PricingAdapter#intimatePricing} to intimate the pricing module of the {@link FeeDetails} being </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>deducted.&lt;/li&gt; &lt;/li&gt;Return boolean value indicating success or failure to the caller module.&lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;/li&gt; &lt;/li&gt;@param feeDetails[] : The details of fees calculated earlier for the given event. See {@link FeeDetails} for more details.&lt;/li&gt; &lt;/li&gt;@return boolean:</p> <ul style="list-style-type: none"> <li>&lt;/li&gt; &lt;/li&gt;true: If the fee is deducted and intimated successfully at the pricing end.&lt;/li&gt;</li> <li>&lt;/li&gt;false: If there is any failure in method execution.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt;</li> <li>&lt;/li&gt;Get the next sub sequence number by calling {@link TransactionAuditItemHelper#getNextSequenceNumber} // As there might be more than one legs to financial transaction.&lt;/li&gt; &lt;/pre&gt; @param feeDetails</li> </ul>
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>&lt;pre&gt;This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. &lt;/pre&gt; @param feeDetails</pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre> This method definition is used to call the actual method implementation in <a href="#">@link PricingAdapter</a> to intimate pricing after a fee has been debited from an account. @param feeDetails </pre>	<pre> //TODO: Such an intimate service is required to be developed at pricing end which will update the various buckets. This method intimates the Pricing module of the charged fee details in the CASA/TD module. This is the context in which this method will be called. For a given event, CASA/TD module will call pricing to calculate the fees to be charged. Pricing will calculate the fees based on Pricing plan and return the same. CASA/TD module has the option to charge the fees, waive them or apply them as it is. The actual action taken on the fees needs to be communicated back to pricing. The details of the fees are passed back to pricing via this method to be updated in the various buckets against the pricing ticket number. See <a href="#">@link PricingDetail#nsfTreatment</a> for more information. The following is the algorithm for the method: Call <a href="#">@link InimationService.intimationService(pricingEventContextDTO: PricingEventContextDTO, transactionReferenceNumber: String, TransactionNature.NORMAL: TransactionNature, transactionDetailDTOs: TransactionDetails[]):boolean</a> Pricing will update the various buckets at its end and return boolean value indicating success or failure. Return this boolean value indicating success or failure to the caller module. The following are the parameters taken by the method along with their significance @param priceTicketNumber : PriceTicketNumber is require to </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>intimate the pricing system. &lt;/li&gt; &lt;/li&gt; &amp;#064;param transactionReferenceNumber : transactionReferenceNumber is require to intimate the pricing system. &lt;/li&gt; &lt;/li&gt; &amp;#064;return boolean: &lt;ul&gt; &lt;/li&gt; &lt;/li&gt; true: If pricing is intimated and the details are successfully updated in pricing module. &lt;/li&gt; &lt;/li&gt; false: If there is any failure in method execution. &lt;/li&gt; &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/li&gt; &lt;/pre&gt; @param priceTicketNumber @param transactionReferenceNumber @return boolean @exception FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimatePricing	<pre>&lt;pre&gt;This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. &lt;/pre&gt; @param feeDetails</pre>	<pre>&lt;pre&gt;This method intimates the Pricing module after a levied fee has been reversed for a transaction. The following is the algorithm for the method: &lt;ol&gt; &lt;/ol&gt; &lt;/li&gt; &lt;/li&gt; Validate the transaction reference number is not null. &lt;/li&gt; &lt;/li&gt; Call pricing module with {@link TransactionNature} as Reversal. &lt;/li&gt; &lt;/li&gt; Pricing will update the counters and return boolean value indicating success or failure. &lt;/li&gt; &lt;/li&gt; Return this boolean value indicating success or failure to the caller. &lt;/li&gt; &lt;/li&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;/li&gt; &lt;/li&gt; &amp;#064;param transactionReferenceNumber : The transaction reference number for which the fees are reversed. &lt;/li&gt; &lt;/li&gt; &amp;#064;return boolean: &lt;ul&gt; &lt;/li&gt; &lt;/li&gt; true: If pricing is intimated and the counters are successfully updated in pricing module. &lt;/li&gt; &lt;/li&gt; false: If there is any failure in method execution. &lt;/li&gt; &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/li&gt; &lt;/pre&gt; @param pricingTicketNumber @param transactionReferenceNumber @param feeRequestInfo</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimatePricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. @param feeDetails </pre>	<pre> This method intimates the Pricing module after a levied fee has been reversed for a transaction. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; Validate the transaction reference number is not null. &lt;/li&gt; &lt;li&gt; Call pricing module with {@link TransactionNature} as Reversal. &lt;/li&gt; &lt;li&gt; Pricing will update the counters and return boolean value indicating success or failure. &lt;/li&gt; &lt;li&gt; Return this boolean value indicating success or failure to the caller. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; transactionReferenceNumber : The transaction reference number for which the fees are reversed. &lt;/li&gt; &lt;li&gt; return boolean: &lt;ul&gt; &lt;li&gt; true: If pricing is intimated and the counters are successfully updated in pricing module. &lt;/li&gt; &lt;li&gt; false: If there is any failure in method execution. &lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;/ul&gt; @param pricingTicketNumber @param transactionReferenceNumber @param feeRequestInfo </pre>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimatePricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. @param feeDetails </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	Method is called in case of System type earmark removed by user. It will intimate pricing that the earmark has been lifted with out debit from the account. @param accountId @param priceTicketNumber @param transactionReferenceNumber @param isSystemInitiated @param holdAmount @param earmarkId @param sessionContext @return
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	intimatePricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. </pre> @param feeDetails	Method is used to lift earmarks. See {@link Earmark} for more info.The method does the following: <ol> <li>Check for mandatory input attributes in the EarmarkDTO object.</li> <li>Call the EarmarkAssembler.toDomainObject() method to get the corresponding domain object</li> <li>Call the EarmarkService.liftHold() method to lift hold on the account</li> <li>Incase of system earmaks lifted by user, intimates the pricing module that the earmark has been removed with out debit in the account.</li> <ol> @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param earmarkDTO MI,{@link EarmarkDTO},earmarkDTO, </br> It contains the following attributes: <ol> <li>key: {@link EarmarkKeyDTO} It should contain only the accountId and earmarkId to be updated.</li> </ol> The rest are not required @return {@link TransactionStatus} </br> <b>Transaction Success</b> <ul> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> </br> <b>Transaction Failure</b> <ul> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> <ul> Error

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Reason may be one of the following <code>&lt;ol&gt;</code> <code>&lt;li&gt;</code> <code>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_NOT_FOUND}&lt;/li&gt;</code> <code>&lt;li&gt;</code> <code>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_ACCOUNT_CLOSED_ERROR}&lt;/li&gt;</code> <code>&lt;li&gt;</code> <code>{@link DemandDepositAccountErrorConstant#DDA_BLANK_INPUT}&lt;/li&gt;</code> <code>&lt;li&gt;</code> <code>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_NO_USER_LIFT_PERMISSION}&lt;/li&gt;</code> <code>&lt;li&gt;</code> <code>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_STATUS_NOT_ACTIVE_ERROR}&lt;/li&gt;</code> <code>&lt;/ol&gt;</code> @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	intimate Pricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. @param feeDetails </pre>	<p>This method intimates the Pricing module of the charged fee details in the CASA, Loan and Settlement mode module.         This is the context in which this method will be called.     For a given event, CASA, Loan and Settlement mode module will call pricing to calculate the fees to be charged.     Pricing will calculate the fees based on Pricing plan and return the same.     CASA, Loan and Settlement mode module has the option to charge the fees, waive them or apply them as it is.     The actual action taken on the fees needs to be communicated back to pricing.     The details of the fees are passed back to pricing via this method to be updated in the various buckets against the pricing ticket number.   </p> <p>Following are the required steps:   1. get the instance of account accounting adapter   {@link com.ofss.fc.app.adapter.impl.account.AccountPricingAdapter}   2. call intimate pricing of adapter   </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>{@link com.ofss.fc.app.adapter.impl.account.AccountPricingAdapter#intimatePricing} &lt;/p&gt; @param priceTicketNumber @param transactionReferenceNumber @param taskModuleType @param accountId @param chargedFeeNode @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimatePricing	<pre>&lt;pre&gt;This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a fee has been debited from an account. &lt;/pre&gt; @param feeDetails</pre>	<p>&lt;p&gt;This method is exposed to external modules to do fees management for CASA, Loan and settlement module.&lt;br&gt; Based on the input, the fees will be debited and then intimated to pricing. &lt;br&gt; In case fee details are not passed as input, then fees will be calculated, debited and intimated to pricing.&lt;br&gt; Following are the required steps:&lt;br&gt; &lt;ol&gt; &lt;li&gt;Fetch pricing details &lt;br&gt; &lt;li&gt;Debit the fee from account.&lt;br&gt; &lt;li&gt;Raise accounting for the charged fees&lt;br&gt; &lt;li&gt;Intimate pricing &lt;br&gt; &lt;/ol&gt; &lt;ol&gt; &lt;li&gt;Fetch pricing details If price ticket number is provided in input than inquire fees based on price ticket number. Price ticket number in input indicates that fees is already inquired by the caller. There could be negotiation performed. The final information is stored into the pricing against this price ticket number. else if pricing details are not provided then inquire fees by the event id. It is not know at this point that whether there is fee linked with this event or not. This call to pricing will return PD with amount, NSFTreatment, discounts and additional fee if there is any. If there is no fee then there will not be any further processing. &lt;br&gt; &lt;li&gt;Debit the fee from account. Debit account can be any module account like CASA, Loan, 2. This method will be responsible to debit the fee from account. &lt;br&gt; &lt;li&gt;Raise accounting for the charged fees&lt;br&gt; &lt;li&gt;Intimate the pricing system by</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>calling  AccountPricingAdapter.intimatePricing(priceTicketNumber,transactionReferenceNumber): boolean  &lt;/ol&gt;  &lt;/p&gt; @throws FatalException @fcb.param  I,String,priceTicketCode,40,, Dialogue ticket obtained from Pricing for fees applicable. If this is supplied, then wrappedPriceFeatureDTO is not required  @fcb.param  I,WrappedPriceFeatureDTO,wrappedPriceFeatureDTO,,,Negotiated fees. If this is supplied, then priceTicketCode is not required @fcb.param  I,String,arrivalId,40, , Indicates the entity for which fees are applicable during the ongoing event. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI, String, accountCode,40 ,, Indicates the account from which the fees have to be funded @fcb.param I, String, eventId, 40,, Indicates the ongoing event for which the fees are applicable on the entity. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,transactionDesc,255, , Transaction description to be logged on the funding account</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>&lt;pre&gt;This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. &lt;/pre&gt; @param pricingTicketNumber</pre>	<pre>&lt;p&gt;This method will intimate pricing for reversal of reward for the original transaction &lt;/p&gt; @param transactionType @param accountTransaction @param amountReversed</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. </pre> @param pricingTicketNumber	This is utility method to intimate pricing based on rewardPricingTicket number.     For ex.In case of sweepout instruction fails, We need to intimate pricing for reversal of reward. @param accountId @param rewardPricingTicketnumber @param originalTransactionReferenceNumber @param eventId
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. </pre> @param pricingTicketNumber	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. </pre> @param pricingTicketNumber	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. @param pricingTicketNumber </pre>	<pre> This method intimates the Pricing module after a levied fee has been reversed for a transaction. The following is the algorithm for the method: <ul style="list-style-type: none"> <li>Validate the transaction reference number is not null.</li> <li>Call pricing module with {@link TransactionNature} as Reversal.</li> <li>Pricing will update the counters and return boolean value indicating success or failure.</li> <li>Return this boolean value indicating success or failure to the caller.</li> </ul> The following are the parameters taken by the method along with their significance <ul style="list-style-type: none"> <li>transactionReferenceNumber: The transaction reference number for which the fees are reversed.</li> <li>return boolean: true: If pricing is intimated and the counters are successfully updated in pricing module. false: If there is any failure in method execution.</li> </ul> @param pricingTicketNumber @param transactionReferenceNumber @param feeRequestInfo </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. @param pricingTicketNumber </pre>	<pre> This method intimates the Pricing module after a levied fee has been reversed for a transaction. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; Validate the transaction reference number is not null. &lt;/li&gt; &lt;li&gt; Call pricing module with {@link TransactionNature} as Reversal. &lt;/li&gt; &lt;li&gt; Pricing will update the counters and return boolean value indicating success or failure. &lt;/li&gt; &lt;li&gt; Return this boolean value indicating success or failure to the caller. &lt;/li&gt; &lt;/ol&gt; The following are the parameters taken by the method along with their significance &lt;ul&gt; &lt;li&gt; transactionReferenceNumber : The transaction reference number for which the fees are reversed. &lt;/li&gt; &lt;li&gt; return boolean: &lt;ul&gt; &lt;li&gt; true: If pricing is intimated and the counters are successfully updated in pricing module. &lt;/li&gt; &lt;li&gt; false: If there is any failure in method execution. &lt;/li&gt; &lt;/ul&gt; &lt;/ul&gt; @param pricingTicketNumber @param transactionReferenceNumber @param feeRequestInfo </pre>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre> This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. @param pricingTicketNumber </pre>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. </pre> @param pricingTicketNumber	<p>This method intimates the Pricing module after a charged fee has been reversed for a transaction. The following is the algorithm for the method: Validate the transaction reference number is not null.  Call account pricing adapter with {@link TransactionNature} as Reversal.  Pricing will update the counters and return boolean value indicating success or failure.  Return this boolean value indicating success or failure to the caller. The following are the parameters taken by the method along with their significance  param transactionReferenceNumber : The transaction reference number for which the fees are reversed. return boolean:  true: If pricing is intimated and the counters are successfully updated in pricing module.  false: If there is any failure in method execution. </p> @param priceTicketNumber @param internalReferenceNo @param depositsFeeRequestInfo
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateReversalToPricing	<pre>This method definition is used to call the actual method implementation in {@link PricingAdapter} to intimate pricing after a levied fee has been reversed for a transaction. </pre> @param pricingTicketNumber	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPriceForAccountClosure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			processingDate @param currencyCodes	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceForAccountClosure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. The method will in turn call the service {@link EventPriceApplicationService#getPriceForAccountClosure} @param sessionContext @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceForAccountClosure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes	This method will be used to post the fees pending for the account including the fee for the closure transaction and update the account balance.  <pre>The method works as follows: <ul><li><li>Check if the input.pricingMap has a pricingTicketNumber for the Periodic fees. </li><li>If the value of pricingTicketNumber is null, then call the service {@link IDDA Pricing Adapter#getPriceForAccountClosure} to get the total periodic fees due from the account. This service will also return the pricing ticket number for periodic fees. </li><li>Call the service {@link FeeManagementService#checkAndLevyFees} with the ticket number to levy the periodic fees. </li><li>Check if the input.pricingMap has a pricingTicketNumber for the Event fees. If yes, the call the service { @link FeeManagementService#checkAndLevyFees} with the ticket number to levy the event fees for account closure. </li></ul></pre> @param accountId @param pricingMap @throws

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceFor Account Closure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes	This method is used to fetch and calculate Pending Periodic Fees for the Account. @param applicationContext @param accountId @param partyId @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceFor Account Closure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes	Method is used to levy pending periodic fees on unclaimed account.   Pricing module will return all periodic fees on prorated basis. For example if calendar monthly fee is 100 INR. and account is becoming unclaimed on 15. Then 50 INR will be charged. @param applicationContext @param demandDepositAccount
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceFor Account Closure	This method will be used for the inquiry of the pending periodic fees for the account at the time of closure. The customer needs to settle all the fees for successful closure. @param pricingContextDTO @param arrangementNumber @param evaluationDate @param processingDate @param currencyCodes	This method will be used to post the fees pending for the account and update the account balance.   @param accountId @param pricingMap @throws FatalException @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	closePricing Plan	This method will be used to unregister or close a pricing plan. If the account is being closed, the pricing plan linked to the account needs to be phased out. @param arrangementNumber @param pricingContractContextDTO	This method will be used to unregister or close a pricing plan. If the account is being closed, the pricing plan linked to the account needs to be phased out.  The method will in turn call the service {@link ContractApplicationService#closePricingPlan} to close the pricing plan. @param arrangementNumber @param pricingContractContextDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	closePricing Plan	This method will be used to unregister or close a pricing plan. If the account is being closed, the pricing plan linked to the account needs to be phased out. @param arrangementNumber @param pricingContractContextDTO	This method is common activity which can be called by any service for activities such as 1.validate & delink bundle benefits from the account to be closed by call to bundle 2.close pricing plan by calling pricing 3.close interest plan by calling interest 4.unregister taxation plan by calling taxation 5.remove all the actions on the account 6.Remove all proportioned OD logged from effective date. 7.delink alternate account. @param closureDTO @param account @param agingAccountStatus @param finalBalance @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	intimatePricingOnSystemHoldRemoval	//TODO Method is called in case of System type earmark removed by user. It will intimate pricing that the earmark has been lifted with out debit from the account. @param accountId @param priceTicketNumber @param transactionReferenceNumber @param isSystemInitiated @param holdAmount @param earmarkId @return	This is an instance variable and not a class variable (static or static final). This is required to support multi-entity wide logging.



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimatePricingOnSystemHoldRemoval	//TODO Method is called in case of System type earmark removed by user. It will intimate pricing that the earmark has been lifted with out debit from the account. @param accountId @param priceTicketNumber @param transactionReferenceNumber @param isSystemInitiated @param holdAmout @param earmarkId @return	Method is used to lift earmarks. See {@link Earmark} for more info.The method does the following: <ol><li>Check for mandatory input attributes in the EarmarkDTO object.</li> <li>Call the EarmarkAssembler.toDomainObject() method to get the corresponding domain object</li> <li>Call the EarmarkService.liftHold() method to lift hold on the account</li> <li>Incase of system earmaks lifted by user, intimates the pricing module that the earmark has been removed with out debit in the account.</li><ol> @param sessionContext MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @param earmarkDTO MI,{@link EarmarkDTO},earmarkDTO, </br> It contains the following attributes: <ol> <li>key:{@link EarmarkKeyDTO} It should contain only the accountId and earmarkId to be updated.</li> </ol> The rest are not required @return {@link TransactionStatus} </br> <b>Transaction Success</b> <ul> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#SUCCESS}</li> </ul> </br> <b>Transaction Failure</b> <ul> <li>{@link com.ofss.fc.infra.enumeration.ResponseCodeType#FATAL_INT_ERROR}</li> </ul> <ul> Error Reason may be one of the following <ol> <li>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_NOT_FOUND}</li> <li>{@link DemandDepositAccountErrorConstant#DDA_EARMARK_ACCOUNT_CLOSED_ERROR}</li> <li>{@link DemandDepositAccountErrorConstant#DDA_BLANK

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>_INPUT}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_EARMRK_NO_USER_LIFT_PERMISSION}&lt;/li&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_EARMRK_STATUS_NOT_ACTIVE_ERROR}&lt;/li&gt; &lt;/ol&gt;  @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>&lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt;</p> <ul style="list-style-type: none"> <li>&lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following paramater &lt;ol&gt;</li> <li>&lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt; Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter &lt;ol&gt; &lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService</p>	<p>This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt; Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter &lt;ol&gt; &lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	@see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}. Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}. Interest rate details contains following information.</p> <ul style="list-style-type: none"> <li>Base rate of interest.- Attached Rate charts can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse specific rate chart to get effective base rate of interest.</li> <li>Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse each rate chart to get cumulative margin rate of interest.</li> <li>UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module.</li> </ul> <p>This method fetches information from pricing module. This method performs following activities.</p> <ul style="list-style-type: none"> <li>Prepares pricingContextDTO of type {@link PricingEventContextDTO}:</li> </ul>	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. In case of offer swap persist rates will be called for all the stages in pricing.</p> <p>@param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method will be used in case of backdated account opening where it would call pricing internally and do the following  ContractApplicationService.computeFeesAndPersistRatesConsideringRepriceEvents(): This method will reprocess all reprice actions like index/rate chart/ppc/fees changes from backdated value date till current posting date. While reprocessing each of this reprice event, pricing will call back following API of CASA:  DDAPricingAccountAdapter.getRepricingConfiguration() with RepriceFormDTO.RepricePurpose=BACKDATED_REPRICE. CASA should fill RepriceFormDTO providing details of reprice. This method will persist rates from back value date till infinity for given stages and return Periodic Fee computation dates. Module should log periodic fee computation actions for these dates. This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following paramater &lt;ol&gt;&lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException</p>	<p>action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}. Interest rate details contains following information.</p> <ul style="list-style-type: none"> <li>Base rate of interest.- Attached Rate charts can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse specific rate chart to get effective base rate of interest.</li> <li>Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse each rate chart to get cumulative margin rate of interest.</li> <li>UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module.</li> </ul> <p>This method fetches information from pricing module. This method performs following activities.</p> <ul style="list-style-type: none"> <li>Prepares pricingContextDTO of type {@link PricingEventContextDTO};</li> </ul>	<p>This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage, od force end, od fixed period, od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate</p> <p>@param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}. Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt; This method fetches information from pricing module. This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO}:</p>	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param sessionContext @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method persist interest rate in amend OD. @param accountId @param partyId @param accountCurrencyCode @param evaluationDate @param arrangementPlanLinkageKeyDTO @return @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method is used to update rate of interest in case of repricing. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>Interest can be thought of as "rent of money". When money is deposited in a bank, interest is typically paid to the depositor as a percentage of the amount deposited. when money is borrowed, interest is typically paid to the lender as a percentage of the amount owed. The percentage of the principal that is paid as a fee over a certain period of time (typically one month or year), is called the interest rate. &lt;br&gt; There are two types of interest in demand deposit module &lt;ol&gt; &lt;li&gt;Credit interest: Bank gives interest on money which is deposited by the customer. Credit interest generally given on {@link EffectiveBalances#clearBalance} &lt;li&gt;Debit interest:Bank levies heavy interest on money which is overdrawn using following facilities &lt;ol&gt; &lt;ol&gt; &lt;li&gt;Overdraft facility: Overdraft facility is product level attribute. Overdraft products are created separately. &lt;li&gt;Overline facility : Please refer {@link AccountFacilities#hasOverline} for more details. &lt;li&gt;Temporary Overdraft: Please refer {@link AccountFacilities#hasTOD} &lt;li&gt;AUF facility : Customer can use funds which are not yet cleared. There is limit present for AUF facility {@link AUFLimit}. &lt;/ol&gt; Please refer following balances for debit interest. &lt;ol&gt; &lt;li&gt; {@link EffectiveBalances#utilizedOverdraftLimit}: &lt;li&gt; {@link EffectiveBalances#overlineAmount} &lt;li&gt; {@link EffectiveBalances#todAmount} &lt;li&gt; {@link EffectiveBalances#utilizedAUFLimit} &lt;/ol&gt; &lt;/ol&gt; &lt;br&gt; Interest rates are maintained in pricing module.At the time of account opening pricing module creates pricing plan for demand deposit account .Interest rates can be tiered or non tiered(absolute). Interest rates are maintained for following types . &lt;ol&gt; &lt;li&gt;Credit interest.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following paramater &lt;ol&gt;&lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException</p>	<p>&lt;li&gt;Overdraft interest &lt;li&gt;Overline interest. &lt;li&gt;TOD(temporary overdraft) interest. &lt;li&gt;AUF(Advance taken against uncleared funds) interest. &lt;/ol&gt;This method is used to inquire interest rates based on {@link InterestRateDetailsDTO}, It provides following types of information about interest rates. &lt;ol&gt; &lt;li&gt;Base rate of interest - &lt;ol&gt; &lt;li&gt;If the base rate defined for an interest type is non-tiered, method will provide the absolute applicable base rate. &lt;li&gt;If the base rate defined for an interest type is tiered, method will give a applicable rate chart. Consumers of this service are expected to analyse rate chart to get the effective base rate of interest. &lt;/ol&gt; &lt;li&gt;Cumulative margin rate of interest - &lt;ol&gt; &lt;li&gt;If the margins defined for an interest type is non-tiered, method will give absolute margin rate (value in BigDecimal). This rate can be positive or nagative. &lt;li&gt;If the margin defined for an interest type is tiered, method will give applicable rate chart. If more than one rate chart are applicable then all rate charts will be returned. Consumers of this service are expected analyse each rate chart to get effective margin rate of interest. &lt;/ol&gt; &lt;li&gt;UDM (User defined margin ) rate of interest- This rate is given account level price benefit chart screen(ALPBC). This rate is always absolute value irrespective of the tiered or non tiered base/margin rates. If UDM is not given for a rate type in ALPBC screen, method will give &lt;code&gt;0 &lt;/code&gt; . &lt;li&gt;Net interest rate : Net rate can be derived using following formula &lt;br&gt; &lt;code&gt; Net rate = base rate + cumulative margin rate + UDM &lt;/code&gt; &lt;br&gt; Consider following scenarios &lt;ol&gt; &lt;li&gt;If base rate and margin rate are non tiered then net rate can be derived. &lt;li&gt;If base rate or margin rate is tiered ,net rate can not be derived.In such scenarios net rate will</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>be null. &lt;/ol&gt; &lt;/ol&gt; Interest rates are fetched from pricing module using adapter layer. This method performs following activities. &lt;ol&gt; &lt;li&gt;Validate demand deposit account: call {@code DemandDepositAccountService#validateDemandDepositAccount}. &lt;li&gt;Call {@link IDDA PricingAdapter#fetchInterestRateDetails} for given {@link InterestRateDetailsDTO}. &lt;li&gt;Prepare response using {@link InterestRateDetailsDTO} returned in step 2. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId, 1..40, The attribute required to fetch balanceStatistics details of DDAccount. @fcb.param MI,CASAInterestTypes,interestRateType,, This parameter captures the types of interest rates like credit,debit. In case of failure -&lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>/This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. &lt;ol&gt; &lt;li&gt;Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt;This method fetches information from pricing module.This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares pricingContextDTO of type {@link PricingEventContextDTO};&lt;br&gt;</p>	<p>This method will be called to create and carry forward the existing UDM to new offer while offer swap. @param demandDepositAccount @param offerSwapDetails @param parentAccountId @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Prepare FactContextDTO : Create instance of BankingFactContextDTO. &lt;li&gt;Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter &lt;ol&gt; &lt;li&gt;pricingContextDTO &lt;li&gt;stage: input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt; &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException</p>	
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	persistAndLogRateChange	<p>This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage</p>	<p>This method will perform the job of linking given od facility to given demandDepositAccount on current posting date. The steps involved in the processing are as follows. &lt;ul&gt; &lt;li&gt;Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.&lt;/li&gt; &lt;li&gt;Now it processes and stamps the common parameters such as</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>and log udm and margin rate            @param accountId @param partyId            @param accountCurrencyCode            @return {@link AccountBalanceRateDTO} @throws FatalException</p>	<p>borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.&lt;/li&gt; &lt;li&gt;After that it fetches the sanctioned limit from facility by using {@link IDDLCAAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCAAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.&lt;/li&gt; &lt;li&gt;Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.&lt;/li&gt; &lt;li&gt;It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper facilities&lt;/li&gt; &lt;li&gt;It calls reapportion for aligning the funds with CASA and LCM.&lt;/li&gt; &lt;li&gt;It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging.&lt;/li&gt; &lt;li&gt;Now it intimates the NPA,Collection module about the linkages.&lt;/li&gt; &lt;/ul&gt; @param demandDepositAccount @param accountLinkage @param isODRateFixed</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistAndLogRateChange	<p>This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od</p>	<p>This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows.            &lt;ul&gt; &lt;li&gt;Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.&lt;/li&gt; &lt;li&gt;Now it processes and stamps the common parameters such as</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>it will log persist rate for that stage and log udm and margin rate            @param accountId @param partyId            @param accountCurrencyCode            @return {@link AccountBalanceRateDTO} @throws FatalException</p>	<p>borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.&lt;/li&gt; &lt;li&gt;Now it finds out the list of {@link BalanceHistoryItem} and updates them with sanctioned OD limit amount and available amount on that date by using {@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)} &lt;li&gt;After that it fetches the sanctioned limit from facility by using {@link IDDLCAAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCAAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.&lt;/li&gt; &lt;li&gt;Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.&lt;/li&gt; &lt;li&gt;It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper facilities&lt;/li&gt; &lt;li&gt;It calls reapportion for aligning the funds with CASA and LCM.&lt;/li&gt; &lt;li&gt;It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging.&lt;/li&gt; &lt;li&gt;Now it intimates the NPA,Collection module about the linkages.&lt;/li&gt; &lt;/ul&gt; @param demandDepositAccount @param accountLinkage @param isODRateFixed</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistAndLogRateChange	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistAndLogRateChange	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistAndLogRateChange	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to link the AUF Facility. @param accountId @param facilityId @param limitAmount @param startDate @param endDate
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistAndLogRateChange	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method works in this flow. <ol> <li>If borrowing entity id in DemandDepositAccount is null then in that case it creates Borrowing entity id for given account and stamps it on account.It ensures account can use limits and collatrels with that borrowing entity id.</li> <li>If start date of borrowing entity is more than current date due to borrowing entity created after cutoff in that case start date is replaced with current date.</li> <li>Then it sets limit on account.LCM has to ensure to call CASA before aliging their funds to ensure proper limit intimation.</li> <li>If temporary excess is null on account then it creates them,then it stamps the sanctioned limit,available limit.</li> <li>If start date of facility is not equal to current date and it is in batch mode then it is case of backdated TE linkage due to holiday processing.In this case it asks pricing for backdated linkage of stage. {@link

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>IDDA PricingAdapter#persistAndLogRateChangeBackdated(String, Date, CASAInterestTypes, boolean)&lt;/li&gt; &lt;li&gt;If start date of facility equals to current then it asks pricing for current dated stage creation. {@link</p> <p>IDDA PricingAdapter#persistAndLogRateChange(String, Date, CASAInterestTypes, boolean)&lt;/li&gt; &lt;li&gt;It first now aligns current state of account by using {@link</p> <p>DemandDepositAccount#reapportionFunds(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)} for non batched linkage and {@link</p> <p>DemandDepositAccount#reapportionFundsForAmountChange(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)} for batched linkages.&lt;/li&gt; &lt;li&gt;Now for history items in range it aligns them by using {@link</p> <p>this#mergeSanctionAmountForHistoryItemsInRange(Date, DemandDepositAccount, Money, ApplicationContext, TransactionAttributes, TransactionDates, com.ofss.fc.enumeration.lcm.limits.FacilityType)&lt;/li&gt;</p> <p>&lt;li&gt;Lastly it calls {@link</p> <p>DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of linkage.&lt;/li&gt; &lt;/ol&gt;</p> <p>@param accountId @param facilityId @param limitAmount @param startDate @param endDate</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetailsForRepricing	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.execute</p>	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param sessionContext @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			Action which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetailsForRepricing	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Discretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of	This method is used to update rate of interest in case of repricing. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.execute Action which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>	
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPeriodic Prices	<p>This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return</p>	
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPeriodic Prices	<p>This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber</p>	<p>This method will be used to perform periodic fee computation for the input arrangement number. It is also responsible to levy the fees on the account if the charging date of the fee falls on the same day. Subsequently the service also logs a computation action in PeriodicFeeLog with the next computation date returned by the pricing service. &lt;br&gt; &lt;pre&gt; The method works as follows: &amp;lt;ul&amp;gt; &amp;lt;li&amp;gt; Call the e-pricing service{@link</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<pre>@param evaluationDate @param computationDate @return</pre>	<pre>IDDA PricingAdapter#getPeriodicPrices} to perform computation with the arrangement number and computation date. This service identifies the periodic fees that are to be computed on the computation date, and performs the computation. It also returns the charging date and the next computation date along with the fee details. &lt;/li&gt; &lt;li&gt; Declare a local array for PeriodicPriceFeatureDTO. &lt;li&gt; Open a loop with the response returned by the pricing service. &lt;li&gt; If the charging date = action date, it implies that the fees have to be levied on the account on the same date. Add the loop record details in the local array for PeriodicPriceFeatureDTO. ElseIf, the charging date is on a future date, then log an action {@link Levy} in PeriodicFeeActionItem for the account number and the date returned by the above service by calling the service {@link PeriodicFeeActionItemService#logPeriodicFeeAction}. &lt;li&gt; &lt;li&gt; Log another action in {@link PeriodicFeeActionItem} for Computation with the computation date returned by the pricing service by calling the service {@link PeriodicFeeActionItemService#logPeriodicFeeAction} &lt;li&gt; &lt;/li&gt; &lt;li&gt; &lt;li&gt; If the size of the array for PeriodicPriceFeatureDTO &gt; 0, call the service {@link FeesManagementService#levyPeriodicFees} to levy the fees. &lt;/li&gt; &lt;/ul&gt; &lt;/pre&gt; @param accountId @param computationDate</pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service { <a href="#">@link EventPriceApplicationService#getPeriodicPrices</a> } @param context @param arrangementNumber @param evaluationDate @param computationDate @return	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service { <a href="#">@link EventPriceApplicationService#getPeriodicPrices</a> } @param context @param arrangementNumber @param evaluationDate @param computationDate @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service { <a href="#">@link EventPriceApplicationService#getPeriodicPrices</a> } @param context @param arrangementNumber @param evaluationDate @param computationDate @return	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service { <a href="#">@link EventPriceApplicationService#getPeriodicPrices</a> } @param context @param arrangementNumber @param evaluationDate @param computationDate @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service { <a href="#">@link EventPriceApplicationService#getPeriodicPrices</a> } @param context @param arrangementNumber @param evaluationDate @param computationDate @return	This method prepares the response using the Account Balances available for the provided Account. @param transactionDates @param demandDepositAccount @param accountBalancesMap @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return	This method adds total fees while closing the account. @param accountId @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return	This method is used to fetch and calculate Pending Periodic Fees for the Account. @param applicationContext @param accountId @param partyId @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getRepriceDatesBetweenDates	This service will be used to fetch the dates on which repricing has taken place on the account between the fromDate and toDate. @param applicationContext @param accountId @param fromDate @param toDate @return @throws FatalException	This service will be used to fetch the dates on which repricing has taken place on the account between the fromDate and toDate. These dates will be used by the backdated processing service, to identify the dates on which the rate change has taken place on the account, and thus interest needs to be recomputed. The service will in turn call the pricing service {@link RepriceStageApplicationService#getRepriceDatesBetweenDates}. Call service {@link AdapterContextHelper#fetchSessionContext()} to retrieve the session Context. @getRepriceDatesBetweenDates @param

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				applicationContext @param accountId @param fromDate @param toDate @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getRatesOnStageWithNetRate	This service will be used to interact between Pricing and DDA and get the net rate applicable for an account for a particular balance component type or stage. @param pricingContextDTO @param arrangementNumber @param stage @param evaluationDate @param accountOpeningDate @return	This service will be used to interact between Pricing and DDA and get the net rate applicable for an account for a particular balance component type or stage. The service will in turn call the pricing service {@link EventRateApplicationService#getRatesOnStageWithNetRate} @param applicationContext @param pricingContextDTO @param arrangementNumber @param stage @param evaluationDate @param accountOpeningDate @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingDetailForTicketNumber	This service is used to compute and return the PriceFeatureDTO. The service is called based on priceTicketNumber. The service in turn calls the pricing service {@link EventPriceApplicationService#getFeesForTicketNumber} @param currencyCodes @param ticketNumber @return @throws FatalException	This method derives the already charged fees for the ticket number. @param eventId @param accountId @return FeeDetailsDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingDetailForTicketNumber	This service is used to compute and return the PriceFeatureDTO. The service is called based on priceTicketNumber. The service in turn calls the pricing service {@link EventPriceApplicationService#getFeesForTicketNumber} @param currencyCodes @param ticketNumber @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPricingDetailForTicketNumber	This service is used to compute and return the PriceFeatureDTO. The service is called based on priceTicketNumber. The service in turn calls the pricing service {@link EventPriceApplicationService#getFeesForTicketNumber} @param currencyCodes @param ticketNumber @return @throws FatalException	This service is used to compute and return the PriceFeatureDTO. The service is called based on priceTicketNumber. The service in turn calls the pricing service {@link EventPriceApplicationService#getFeesForTicketNumber} @param currencyCodes @param ticketNumber @return @throws FatalException
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getComputedPeriodicFees	This service will be used to return the already computed periodic fees from pricing. As part of the computation action, the periodic fees is computed and stored in e-pricing. During the execution of the action PeriodicFeecharging, this computed fee has to be retrieved from e-pricing and levied on the customer account. This service returns the computed fee amount which has to be levied on the account. The service in turn calls the service {@link EventPriceApplicationService#getComputedPeriodicFees} @param context @param arrangementNumber @param evaluationDate @param computationDate @param partyId @param accountCurrency @param chargingDate @return	This method will build the wrapped price feature dto for the amount to be charged as part of bundle maintenance fee.   If the {@link com.ofss.fc.app.pricing.dto.price.PeriodicPriceFeatureDTO} is supplied, the {@link com.ofss.fc.app.pricing.dto.price.WrappedPriceFeatureDTO} is built out of it, else the computed prices are inquired from pricing and the dto is populated. @param applicationContext @param partyBundleLinkage @param accountCurrency @param partyId @param periodicPriceFeatureDTOArray @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getComputedPeriodicFees	<p>This service will be used to return the already computed periodic fees from pricing. As part of the computation action, the periodic fees is computed and stored in e-pricing. During the execution of the action PeriodicFeecharging, this computed fee has to be retrieved from e-pricing and levied on the customer account. This service returns the computed fee amount which has to be levied on the account. The service in turn calls the service <a href="#">EventPriceApplicationService#getComputedPeriodicFees</a> @param context @param arrangementNumber @param evaluationDate @param computationDate @param partyId @param accountCurrency @param chargingDate @return</p>	<p>This method will build the wrapped price feature dto for the amount to be charged as part of bundle maintenance fee. &lt;br/&gt; If the <a href="#">com.ofss.fc.app.pricing.dto.price.PeriodicPriceFeatureDTO</a> is supplied, the <a href="#">com.ofss.fc.app.pricing.dto.price.WrappedPriceFeatureDTO</a> is built out of it, else the computed prices are inquired from pricing and the dto is populated. @param applicationContext @param partyBundleLinkage @param accountCurrency @param partyId @param periodicPriceFeatureDTOArray @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getComputedPeriodicFees	<p>This service will be used to return the already computed periodic fees from pricing. As part of the computation action, the periodic fees is computed and stored in e-pricing. During the execution of the action PeriodicFeecharging, this computed fee has to be retrieved from e-pricing and levied on the customer account. This service returns the computed fee amount which has to be levied on the account. The service in turn calls the service {<a href="#">@link</a> EventPriceApplicationService#getComputedPeriodicFees} @param context @param arrangementNumber @param evaluationDate @param computationDate @param partyId @param accountCurrency @param chargingDate @return</p>	<p>@param applicationContext @param partyBundleLinkage @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getComputedPeriodicFees	<p>This service will be used to return the already computed periodic fees from pricing. As part of the computation action, the periodic fees is computed and stored in e-pricing. During the execution of the action PeriodicFeecharging, this computed fee has to be retrieved from e-pricing and levied on the customer account. This service returns the computed fee amount which has to be levied on the account. The service in turn calls the service {<a href="#">@link</a> EventPriceApplicationService#getComputedPeriodicFees} @param context @param arrangementNumber @param evaluationDate @param computationDate @param partyId @param accountCurrency @param chargingDate @return</p>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getComputedPeriodicFees	This service will be used to return the already computed periodic fees from pricing. As part of the computation action, the periodic fees is computed and stored in e-pricing. During the execution of the action PeriodicFeecharging, this computed fee has to be retrieved from e-pricing and levied on the customer account. This service returns the computed fee amount which has to be levied on the account. The service in turn calls the service <a href="#">EventPriceApplicationService#getComputedPeriodicFees</a> @param context @param arrangementNumber @param evaluationDate @param computationDate @param partyId @param accountCurrency @param chargingDate @return	This method will be used to levy period fees on the account on the fees charging date as intimated by e-pricing. This service will levy the fees on the main account or the alternate account if one has been maintained for fees posting.   It is assumed that when the service is called, the periodic fees to be charged on the input.chargingDate has already been computed and stored in e-pricing. Hence, it calls the service to retrieve the computed price and subsequently levies the fees on the account.   <pre> It works as follows : <ul><li> Call the service <a href="#">IDDA PricingAdapter#getComputedPeriodicFees</a> to get the calculated periodic fees. </li><li> Call the service <a href="#">@FeesManagementService#levyPeriodicFees</a> with the periodic fee details returned by the above service to levy the fees on the account. </li></ul> @param applicationContext @param accountId @param chargingDate
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service <a href="#">EventPriceApplicationService#getPeriodicPrices</a> @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service <a href="#">{@link EventPriceApplicationService#getPeriodicPrices}</a> @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	This method will be used to perform periodic fee computation for the input arrangement number. It is also responsible to levy the fees on the account if the charging date of the fee falls on the same day. Subsequently the service also logs a computation action in PeriodicFeeLog with the next computation date returned by the pricing service. <pre> The method works as follows: <ul style="list-style-type: none"> <li>&amp;lt;li&amp;gt; Call the e-pricing service <a href="#">{@link IDDA PricingAdapter#getPeriodicPrices}</a> to perform computation with the arrangement number and computation date. This service identifies the periodic fees that are to be computed on the computation date, and performs the computation. It also returns the charging date and the next computation date along with the fee details. &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Declare a local array for PeriodicPriceFeatureDTO. &amp;lt;/li&amp;gt; Open a loop with the response returned by the pricing service. &amp;lt;/li&amp;gt; If the charging date = action date, it implies that the fees have to be levied on the account on the same date. Add the loop record details in the local array for PeriodicPriceFeatureDTO. ElseIf, the charging date is on a future date, then log an action <a href="#">{@link Levy}</a> in PeriodicFeeActionItem for the account number and the date returned by the above service by calling the service <a href="#">{@link PeriodicFeeActionItemService#logPeriodicFeeAction}</a>. &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Log another action in <a href="#">{@link PeriodicFeeActionItem}</a> for Computation with the computation date returned by the pricing service by calling the service <a href="#">{@link PeriodicFeeActionItemService#logPeriodicFeeAction}</a> &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; If the size of the array for PeriodicPriceFeatureDTO &amp;gt; 0, call the service <a href="#">{@link</a></li></ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				FeesManagementService#levyPeriodicFees} to levy the fees. </li> </ul> </pre> @param accountId @param computationDate
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	This method prepares the response using the Account Balances available for the provided Account. @param transactionDates @param demandDepositAccount @param accountBalancesMap @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @param priceTicketNumber @return	This method adds total fees while closing the account. @param accountId @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPeriodicPrices	This service is used to compute and return the periodic fees. The service is called as part of the periodic fees computation action on the fee computation date. The service in turn calls the pricing service {@link EventPriceApplicationService#getPeriodicPrices} @param context @param arrangementNumber @param evaluationDate @param computationDate @param	This method is used to fetch and calculate Pending Periodic Fees for the Account. @param applicationContext @param accountId @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			priceTicketNumber @return	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to inquire interest rate details for given {@link CASAInterestTypes}.Interest rate details contains following information. <ol> <li>Base rate of interest.- Attached Rate charts can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse specific rate chart to get effective base rate of interest. <li>Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered.For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details.Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. <li>UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. </ol>This method fetches information from pricing module.This method performs following activities. <ol> <li>Prepares pricingContextDTO of type {@link PricingEventContextDTO};  Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service:

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>EventRateApplicationService#getRatesWithLenientComputation following parameter &lt;ol&gt;  &lt;li&gt;pricingContextDTO &lt;li&gt;stage:  input.interestRateType &lt;li&gt;evaluationDate : Current Process Date.(Get it from common service) &lt;/ol&gt;  &lt;br&gt; &lt;li&gt;Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula. &lt;br&gt; &lt;code&gt;Net rate = base rate + cumulative interest rate + UDM&lt;/code&gt; @see DDAAInquiryApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Discretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Discretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. In case of offer swap persist rates will be called for all the stages in pricing. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be used in case of backdated account opening where it would call pricing internally and do the following ContractApplicationService.computeFeesAndPersistRatesConsideringRepriceEvents(): This method will reprocess all reprice actions like index/rate chart/ppc/fees changes from backdated value date till current posting date. While reprocessing each of this reprice event, pricing will call back following API of CASA: DDAPricingAccountAdapter.getRepricingConfiguration() with RepriceFormDTO.RepricePurpose=BACKDATED_REPRICE. CASA should fill RepriceFormDTO providing details of reprice. This method will persist rates from back value date till infinity for given stages and return Periodic Fee computation dates. Module should log periodic fee computation actions for these dates. This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param sessionContext @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method persist interest rate in amend OD. @param accountId @param partyId @param accountCurrencyCode @param evaluationDate @param arrangementPlanLinkageKeyDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to update rate of interest in case of repricing. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	fetchInterestRateDetails	<p>This method will fetch the interest rate details that will be required for updating rate in Demand deposit account</p> <p>@param accountId  @param partyId  @param accountCurrencyCode  @return {@link AccountBalanceRateDTO}  @throws FatalException</p>	<p>Interest can be thought of as "rent of money". When money is deposited in a bank, interest is typically paid to the depositor as a percentage of the amount deposited. when money is borrowed, interest is typically paid to the lender as a percentage of the amount owed. The percentage of the principal that is paid as a fee over a certain period of time (typically one month or year), is called the interest rate.</p> <p>There are two types of interest in demand deposit module</p> <ul style="list-style-type: none"> <li>Credit interest: Bank gives interest on money which is deposited by the customer. Credit interest generally given on {@link EffectiveBalances#clearBalance}</li> <li>Debit interest: Bank levies heavy interest on money which is overdrawn using following facilities</li> </ul> <ul style="list-style-type: none"> <li>Overdraft facility: Overdraft facility is product level attribute. Overdraft products are created separately.</li> <li>Overline facility : Please refer {@link AccountFacilities#hasOverline} for more details.</li> <li>Temporary Overdraft: Please refer {@link AccountFacilities#hasTOD}</li> <li>AUF facility : Customer can use funds which are not yet cleared. There is limit present for AUF facility {@link AUFLimit}. Please refer following balances for debit interest.</li> </ul> <ul style="list-style-type: none"> <li>{@link EffectiveBalances#utilizedOverdraftLimit}: </li> <li>{@link EffectiveBalances#overlineAmount}</li> <li>{@link EffectiveBalances#todAmount}</li> <li>{@link EffectiveBalances#utilizedAUFLimit}</li> </ul> <p>Interest rates are maintained in pricing module. At the time of account opening pricing module creates pricing plan for demand deposit account .Interest rates can be tiered or non tiered(absolute). Interest rates are maintained for following types .</p> <ul style="list-style-type: none"> <li>Credit interest.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> &lt;li&gt;Overdraft interest &lt;/li&gt; &lt;li&gt;Overline interest. &lt;/li&gt; &lt;li&gt;TOD(temporary overdraft) interest. &lt;/li&gt; &lt;li&gt;AUF( Advance taken against uncleared funds) interest. &lt;/li&gt; &lt;/ol&gt; This method is used to inquire interest rates based on {<a href="#">@link InterestRateDetailsDTO</a>}, It provides following types of information about interest rates. &lt;ol&gt; &lt;li&gt;Base rate of interest - &lt;/li&gt; &lt;li&gt;If the base rate defined for an interest type is non-tiered, method will provide the absolute applicable base rate. &lt;/li&gt; &lt;li&gt;If the base rate defined for an interest type is tiered, method will give a applicable rate chart. Consumers of this service are expected to analyse rate chart to get the effective base rate of interest. &lt;/li&gt; &lt;li&gt;Cumulative margin rate of interest - &lt;/li&gt; &lt;li&gt;If the margins defined for an interest type is non-tiered, method will give absolute margin rate (value in BigDecimal). This rate can be positive or nagative. &lt;/li&gt; &lt;li&gt;If the margin defined for an interest type is tiered, method will give applicable rate chart. If more than one rate chart are applicable then all rate charts will be returned. Consumers of this service are expected analyse each rate chart to get effective margin rate of interest. &lt;/li&gt; &lt;li&gt;UDM (User defined margin ) rate of interest- This rate is given account level price benefit chart screen(ALPBC). This rate is always absolute value irrespective of the tiered or non tiered base/margin rates. If UDM is not given for a rate type in ALPBC screen, method will give &lt;code&gt;0 &lt;/code&gt; . &lt;/li&gt; &lt;li&gt;Net interest rate : Net rate can be derived using following formula &lt;br&gt; &lt;code&gt; Net rate = base rate + cumulative margin rate + UDM &lt;/code&gt; &lt;br&gt; Consider following scenarios &lt;ol&gt; &lt;li&gt;If base rate and margin rate are non tiered then net rate can be derived. &lt;/li&gt; &lt;li&gt;If base rate or margin rate is tiered ,net rate can not be derived.In such scenarios net rate will </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>be null. &lt;/ol&gt; &lt;/ol&gt; Interest rates are fetched from pricing module using adapter layer. This method performs following activities. &lt;ol&gt; &lt;li&gt;Validate demand deposit account: call {@code DemandDepositAccountService#validateDemandDepositAccount}. &lt;li&gt;Call {@link IDDA PricingAdapter#fetchInterestRateDetails} for given {@link InterestRateDetailsDTO}. &lt;li&gt;Prepare response using {@link InterestRateDetailsDTO} returned in step 2. &lt;/ol&gt; @fcb.param MI,SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,accountId, 1..40, The attribute required to fetch balanceStatistics details of DDAccount. @fcb.param MI,CASAInterestTypes,interestRateType,, This parameter captures the types of interest rates like credit,debit. In case of failure -&lt;br&gt; &lt;b&gt;Transaction Failure&lt;/b&gt;&lt;br&gt; Error Reason may be one of the following:&lt;br&gt; &lt;ul&gt; &lt;li&gt;{@link DemandDepositAccountErrorConstant#DDA_ACCOUNT_NOT_PRESENT}&lt;/li&gt; &lt;/ul&gt; @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetails	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException	This method will be called to create and carry forward the existing UDM to new offer while offer swap. @param demandDepositAccount @param offerSwapDetails @param parentAccountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetailsForOfferSwap		This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when magin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. In case of offer swap persist rates will be called for all the stages in pricing. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetailsForOfferSwap		This method persist interest rate in amend OD. @param accountId @param partyId @param accountCurrencyCode @param evaluationDate @param arrangementPlanLinkageKeyDTO @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchInterestRateDetailsForOfferSwap		This method will be called to create and carry forward the existing UDM to new offer while offer swap. @param demandDepositAccount @param offerSwapDetails @param parentAccountId @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	inquireRepriceInformation	@param arrangementNumber @param processDate @return	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	inquireRepriceInformation	@param arrangementNumber @param processDate @return	This method will fetch the repricing information for the given account. @param sessionContext @param accountId @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateRefundFeeDetails	This method will be used to update the transaction details FlattenedFeeTransactionDetails with the reversed amounts. IT will update the refunded amount field of FeeChargeDetail corresponding to the base price id mentioned in the DTO {@link ReversalDetailsDTO} @param reversalDetailsDTOs @throws FatalException	This method will process the reverse fees which will be refunded to the corresponding CASA account and the accounting entries will be passed.  Also Pricing will be intimated about the refund transaction and the corresponding details will be updated   @param sessionContext @param accountNo @param reversalDetailsDTO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	repriceDDADetails	This method is used to reprice the pricing details for an account number	Re-prices the regular and penalty base interest index offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing interest index offerings, re-evaluating the regular and penalty interest policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the regular and penalty margin index offerings in the {@link



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing margin offerings, re-evaluating the regular and penalty margin policy charts at all levels and merging them to create new offerings. Campaign variances, if any, are re-fetched and re-applied to the new offerings. Account-level variances are retained. Re-prices the price offerings in the {@link PricingPlan} associated with the {@code arrangementNumber}. This involves discarding all existing price offerings, re-evaluating the price policy charts at all levels and merging them to create new price offerings. Campaign variances, if any, are re-fetched and re-applied to the new price offerings. Account-level variances are discarded with the earlier price offerings, and are NOT reinstated with the new price offerings.</p> <p>The {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext) TicketService.startDialogue(SessionContext)} method MUST be called to acquire a {@link PricingTicketDTO} before calling this method. The {@link PricingTicketDTO} must then be passed as a parameter to all subsequent calls to this method within the same dialogue. If no ticket or an invalid ticket is passed in, the method simply throws a {@link FatalException} and aborts.</p> <p>@param sessionContext the session context @param dDA PricingPlanAdapterDTO @throws FatalException if the input is invalid, or there is a problem while repricing</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	repriceDDADetails	This method is used to reprice the pricing details for an account number	This method uses DDAPricingAdapter for repricing the pricing details for the given accountId as the input which reprices the following data at pricing module: repriceInterestIndexOfferings repriceMarginIndexOfferings repricePriceOfferings @param demandDepositAccount @param repriceDate
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	This method is responsible for levying Periodic Fees on the account in case bulking flag is true.   Bulking essentially means for all the fees associated with an account the debit (i.e deduction of the fee amount) will be cumulated.   The bulking indicator and the frequency is setup in pricing module.   Once the fee details are populated by the caller method this method internally gives a call   to levyGiven which in turn will debit the account and intimate pricing.   @param depositsFeeRequestInfo @param accountTransaction @param feeAmountsToBeCharged @param isBalanceLimitCrossed @param pricingDetails @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	<p>This method is exposed to external modules to do fees management for CASA, Loan and settlement module.  Based on the input, the fees will be debited and then intimated to pricing.   In case fee details are not passed as input, then fees will be calculated, debited and intimated to pricing.  Following are the required steps:  <ol> <li>Fetch pricing details   <li>Debit the fee from account.  <li>Raise accounting for the charged fees  <li>Intimate pricing   </ol> <ol> <li>Fetch pricing details If price ticket number is provided in input than inquire fees based on price ticket number. Price ticket number in input indicates that fees is already inquired by the caller. There could be negotiation performed. The final information is stored into the pricing against this price ticket number. else if pricing details are not provided then inquire fees by the event id. It is not know at this point that whether there is fee linked with this event or not. This call to pricing will return PD with amount, NSFTreatment, discounts and additional fee if there is any. If there is no fee then there will not be any further processing.   <li>Debit the fee from account. Debit account can be any module account like CASA, Loan, 2. This method will be responsible to debit the fee from account.   <li>Raise accounting for the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>charged fees&lt;br&gt; &lt;li&gt;Intimate the pricing system by calling  AccountPricingAdapter.intimatePricing(priceTicketNumber,transactionReferenceNumber): boolean &lt;/ol&gt;  &lt;/p&gt; @throws FatalException @fcb.param I,String,priceTicketCode,40,, Dialogue ticket obtained from Pricing for fees applicable. If this is supplied, then wrappedPriceFeatureDTO is not required @fcb.param I,WrappedPriceFeatureDTO,wrappedPriceFeatureDTO,,,Negotiated fees. If this is supplied, then priceTicketCode is not required @fcb.param I,String,arrangementId,40, , Indicates the entity for which fees are applicable during the ongoing event. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,accountCode,40,, Indicates the account from which the fees have to be funded @fcb.param I,String,eventId,40,, Indicates the ongoing event for which the fees are applicable on the entity. If either priceTicketCode or wrappedPriceFeatureDTO is supplied, then this is not required. @fcb.param MI,String,transactionDesc,255, , Transaction description to be logged on the funding account</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateFeeDetailsForTicketNumber	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createUDM Definition	This method will be used to create UDM definition for given account.  . @param accountId @param domain @param domainCategory @param eventRateVarianceDTOs	This method will be used to execute bonus interest processing.  Sequence for execution-  <ol><li>Inquire product id by loading DDAccount.<li>check for UDM maintenance at product level if present then <li>Get all the facts from consulting to evaluate the rule. <li>Validate rule for bonus interest eligibility and if account is eligible for bonus interest then <li>log the non financial statement item with description. <li>check if bonus interest is applicable for current cap cycle then call Interest module to get the start date and end date of current cycle and Call pricing to create back dated UDM Definition at account level, provide start date and end date also.  and instead of logging backdated action process backdated right away. <li>And if bonus interest is applicable for next cap cycle then create future dated UDM with start and end date and log the rate change action with due date as start date. </ol> @param applicationContext Application Context which will be set in batch process. @param actionKey @param actionRequest every subclass need to cast this to appropriate subclass of {@link ActionKey} @param dueDate due date for the action
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createUDM Definition	This method will be used to create UDM definition for given account.  . @param accountId @param domain @param domainCategory @param eventRateVarianceDTOs	This method will be used to create UDM definition for given account.  A call will be made to pricing module through {@link IDDA PricingAdapter}. This method will be used to create backdated or future dated UDM depending upon capitalization dates given. @param applicationContext @param demandDepositAccount @param bonusInterestRateDetails @param capStartDate @param capEndDate @param capitalizationCycle TODO @return udmEffectiveStartDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createUDM Definition	This method will be used to create UDM definition for given account.  . @param accountId @param domain @param domainCategory @param eventRateVarianceDTOs	This method will be used to create UDM definition for given account.  @param accountId @param domain @param domainCategory @param eventRateVarianceDTOs
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardDate	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @param partyId @param currencyCodes @param transactionAmountInTransactionCCy @param transactionAmountInAccountCcy @return @throws FatalException	This method will make a call to pricing during any financial transaction with the event id as input and fetch reward due date if reward is applicable to this event and also log an end of day action. @param accountId @param eventId @param transactionAmountInTransactionCCy @param txnRefNo
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardDate	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @param partyId @param currencyCodes @param transactionAmountInTransactionCCy @param transactionAmountInAccountCcy @return @throws FatalException	If the transaction being executed is eligible for reward, this method will make a call to pricing and fetch the reward due date. @param accountId @param eventId @return Date @throws FatalException @param accountId @param eventId @param partyId @param currencyCodes @param transactionAmountInTransactionCcy @param transactionAmountInAccountCcy @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardDate	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @param partyId @param currencyCodes @param transactionAmountInTransactionCCy @param transactionAmountInAccountCcy @return @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardDate	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @param partyId @param currencyCodes @param transactionAmountInTransactionCCy @param transactionAmountInAccountCcy @return @throws FatalException	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardAmountAndPricingDetail	When the current posting date is same as the reward due date and the end of day batch is running, this method will be called with account id, account currency and reward due date as input to fetch the reward amount and the tax on reward due to the customer. @param accountId @param accountCurrency @param dueDate @return WrappedPriceFeatureDTO	This method will delegate call to the corresponding executor for processing of the logged action.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardAmountAndPricingDetail	When the current posting date is same as the reward due date and the end of day batch is running, this method will be called with account id, account currency and reward due date as input to fetch the reward amount and the tax on reward due to the customer. @param accountId @param accountCurrency @param dueDate @return WrappedPriceFeatureDTO	This method makes a call to pricing with account id, account currency and due date as inputs to fetch the total reward amount due on this date. @param dueDate @param accountId @param demandDepositAccount @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardAmountAndPricingDetail	When the current posting date is same as the reward due date and the end of day batch is running, this method will be called with account id, account currency and reward due date as input to fetch the reward amount and the tax on reward due to the customer. @param accountId @param accountCurrency @param dueDate @return WrappedPriceFeatureDTO	Method to credit Reward Amount On Closure @param sessionContext @param accountId @param accountTransaction @param demandDepositAccountService @param ddaPricingAdapter
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRewardAmountAndPricingDetail	When the current posting date is same as the reward due date and the end of day batch is running, this method will be called with account id, account currency and reward due date as input to fetch the reward amount and the tax on reward due to the customer. @param accountId @param accountCurrency @param dueDate @return WrappedPriceFeatureDTO	When the current posting date is same as the reward due date and the end of day batch is running, this method will be called with account id, account currency and reward due date as input to fetch the reward amount and the tax on reward due to the customer. @param accountId @param accountCurrency @param dueDate @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method is used by ELCM to intimate DDA about the OD DeLinkage before Expiry. The method accepts the accountId and accountFacilityLinkage. @param sessionContext @param accountId @param accountFacilityLinkage @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method is used by ELCM to intimate DDA about the OD DeLinkage before Expiry. The method accepts the accountId and accountFacilityLinkage. @param sessionContext @param accountId @param accountFacilityLinkage @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method is used by ELCM to intimate DDA about the OD DeLinkage before Expiry. The method accepts the accountId and accountFacilityLinkage. @param sessionContext @param accountId @param accountFacilityLinkage @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when magin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when magin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. In case of offer swap persist rates will be called for all the stages in pricing. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method will be used in case of backdated account opening where it would call pricing internally and do the following ContractApplicationService.computeFeesAndPersistRatesConsideringRepriceEvents(): This method will reprocess all reprice actions like index/rate chart/ppc/fees changes from backdated value date till current posting date. While reprocessing each of this reprice event, pricing will call back following API of CASA: DDAPricingAccountAdapter.getRepricingConfiguration() with RepriceFormDTO.RepricePurpose=BACKDATED_REPRICE. CASA should fill RepriceFormDTO providing details of reprice. This method will persist rates from back value date till infinity for given stages and return Periodic Fee computation dates. Module should log periodic fee computation actions for these dates. This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is getting expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage,od force end,od fixed period,od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's.
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method persist interest rate in amend OD. @param accountId @param partyId @param accountCurrencyCode @param evaluationDate @param arrangementPlanLinkageKeyDTO @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	persistRates	This method will be used to persist the rates. @param arrangementNumber @param effectiveDate @return	This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's.
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchWrapperFromEvent	This method will be used to fetch wrapper from the event @param pricingContextDTO @param accountId @param rateTypes @param evaluationDate @return	@param sessionContext @param pricingContextDTO @param accountId @param rateTypes @param evaluationDate @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	On account opening, method id called to mark the fees as deferred or capitalised. Following steps are performed. <ol> <li>Get the outlay funds which are stored against recommendation pricing ticket.Refer DDAOpeningApplicationService.inquireServicingFeesForRecommendation. <li>Mark fees as deferred or capitalised in pricing module. <li>Store outlay funds against account number. </ol> @param pricingTicketNumber @param accountId @param productId
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	On Amend OD workflow, After inquiring fees we mark the fees as cap or deferred and update account id for fee ticket number. @param odAmendConfigurationDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	Method to credit Reward Amount On Closure @param sessionContext @param accountId @param accountTransaction @param demandDepositAccountService @param ddaPricingAdapter
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method gives periodic fee ticket numbers for a particular process date. @param accountId @param originalTransactionReferneceNumber @param processDate @return
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will check and release all the holds that are being made on the account. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This service will be used to levy periodic fees on demandDepositAccount. In case the bulkingRequired indicator is set to True in e-pricing, the service will consolidate the periodic fee amount due for the date and levy the same. It will also update the periodicFeeAmount at the account level which will subsequently be used for interest computation. <pre>The method works as follows: <ul><li><li> Open a loop for the input.periodicPriceFeatureDTO <ul><li><li> If the value of the input parameter input.isBulkingRequired = True, add the value of the fee amount in a local variable.</li><li><li> Prepare the object DDAFeeRequestInfo with the input details {In case of bulking, the FeeDetails will have only 1 row, else there will be a separate row for each fee code}</li><li><li> Call the service {@this.checkAndlevyFees} with the DDAFeeRequestInfo prepared above. </li><li><li> Open a loop for the FeeAmount in the return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>parameter of the above service. &lt;li&gt; Add the actually debited fee amount in a local variable. &lt;/li&gt; &lt;li&gt; &lt;/li&gt; &lt;li&gt; Call {@link AlternateAccountService#fetchAlternateAccountIdForFeesPosting()} to get the alternate account id.&lt;/li&gt; &lt;li&gt; If the alternate account exists, then load the alternate account, else load the input.account by calling the service {@link DemandDepositAccountRepository#load}&amp;lt;/li&gt; &lt;li&gt; If alternate account exists, then call the service {@link DemandDepositAccount#increasePeriodicFeeAmount} for the alternate account, else call the service for the main account with the total debited fee amount calculated above. &lt;/li&gt; &lt;li&gt; Call the service {@link DemandDepositAccountRepository#update} to update the new periodicFeeAmount at the account level. &lt;/li&gt; &lt;li&gt; Raise an accounting event by calling the service {@link IDomainAccountingAdapter#raiseAccountingEvent}&amp;lt;/li&gt; &lt;li&gt; Log an action {@link BalanceChangeAction} with the debited account, i.e. the alternate account or the main account as the case may be. &lt;/pre&gt; @param listOfPriceFeatureDTOs @param accountTransaction @param depositsFeeRequestInfo @param pricingTicketNumber</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will return all the periodic fee action item for the account id and transaction reference number. @param accountId @param actionType @param actionDate @param transactionReferenceNumber @param pricingTicketNumber @return List<PeriodicFeeActionItemDTO>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This service will fetch the pricing information that is interest rate and margin rate and the servicing charges if applicable for the demand deposit product. The pricing details are fetched from the Pricing module. Any call to the pricing engine must be initiated by starting a dialogue. The pricing engine returns a pricing ticket number which is the unique identifier for the account configuration. Separate calls for fetching interest rates, margin rates and servicing charges are then done to the pricing engine using the same pricing ticket number.  Refactored to this module from (@link DemandDepositSimulationApplicationService} on March 27. @fcb.param MI,SessionContext,sessionContext, Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,ApplicationParameterDTO, applicationParameterDTO, , Indicates the extracted application details required for making a call to the pricing module @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate. @throws FatalException @exception</p> <p>&lt;br&gt; {@link DemandDepositAccountErrorConstant#INTEREST_OFFERINGS_NOT_FOUND} : FC_OR_PR_0001 &lt;br&gt; This error occurs when:&lt;br&gt; Interest rate offerings cannot be fetched from Pricing module.&lt;br&gt;</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	<p>This service will fetch the net interest rate for demand deposit product details that are passed. The net interest rate is fetched from Pricing module.&lt;br&gt; Refactored to this module from {@link DemandDepositSimulationApplicationService} on March 27. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,DemandDepositProductGroupLinkageDTO, demandDepositProductGroupLinkageDTO, , Indicates the product group details for the account configuration @fcb.param MI,ApplicantDTO, primaryApplicantDTO, , Indicates the information that is linked to the primary applicant of the application @fcb.param MI, DDAccountConfigurationDTO, ddAccountConfigurationDTO, , Indicates the information linked to the account configuration for which pricing information is being fetched. @return NetInterestRateResponse, , Indicates the response class that containing information such as pricing ticket number unique for each account configuration and net interest rate.</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method is responsible for logging the Periodic Fee Charging action if the charging date is after the computation date. In addition to this the ProcessedBasePriceDTO that has the charge date after the computation date is removed from the processedBasePriceDTOList. Thus on the completion of this method processedBasePriceDTOList will have the details of the fees that has the computation date equal to the charge date which has to be charged. @param accountId @param computationDate @param priceTicketNumber @param processedBasePriceDTOList
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will be used to levy period fees on the account on the fees charging date as intimated by e-pricing. This service will levy the fees on the main account or the alternate account if one has been maintained for fees posting.   It is assumed that when the service is called, the periodic fees to be charged on the input.chargingDate has already been computed and stored in e-pricing. Hence, it calls the service to retrieve the computed price and subsequently levies the fees on the account.   <pre> It works as follows : <ul><li><li> Call the service {@link IDDA PricingAdapter#getComputedPeriodicFees} to get the calculated periodic fees. </li><li> Call the service {@FeesManagementService#levyPeriodicFees} with the periodic fee details returned by the above service to levy the fees on the account. </li></ul></pre> @param applicationContext @param accountId @param chargingDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will populate the pricing details out of price feature DTO. This method has been extracted as there cannot be multiple @param priceFeatureDTO @param accountCurrencyCode @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	Method prepares and stores servicing fee response. @param accountServicingFeeRequestDTO @param priceFeaturesDTOs @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	Prepare fee request from {@link ODAccountConfigurationParameterDTO}. Important step is to include preparation of inputs facts. @param accountConfigurationParameterDTO @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will fetch PricingTicketNumber and generate recommendation. @param offerSwapDetails @return String @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPricingTicketNumber	This method will fetch PricingTicketNumber for pricing plan. @return @throws FatalException	Method prepares and stores servicing fee response. @param accountServicingFeeRequestDTO @param priceFeaturesDTOs @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateBundleBenefitApplicationStatus	This method updates Bundle Benefits in pricing @param arrangementNumber @param benefitApplicabilityDTOArray	When recommendation is created, we need to update bundle benefit status as well. Bundle module stores benefit status is string format. For ex.Fixed:All,Variable:All @param priceTicketNumber @param demandDepositAccount @param postingDate @param amendConfigurationDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateBundleBenefitApplicationStatus	This method updates Bundle Benefits in pricing @param arrangementNumber @param benefitApplicabilityDTOArray	Method to update bundle benefits in pricing @param arrangementNumber @param benefitApplicabilityDTOArray
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	getPriceFeatureDTOOnEvent		
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchDeferredFeesForArrangementNumber	This method is to get Differed Fees for the nominated account. @param currencyCodes @param arrangementNumber @return	This method handles differed fee handling for given casa account.It fetches the differed fees from pricing and then charges it using fee helper. @param accountId @param debitInstruction @param applicationContext @param feeHelper @param ddaPricingAdapter @param ccyCodes @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchDeferredFeesForArrangementNumber	This method is to get Differed Fees for the nominated account. @param currencyCodes @param arrangementNumber @return	Levy fees which are deferred. Fee Treatment is stored at preferences. in product module @param accountId @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	generatePricingContractContextDTO	This method would generate the pricing contract context dto using the demand deposit account as input which would then be used in re-pricing @param idemandDepositAccount @return	This method would generate the pricing contract context dto using the demand deposit account as input which would then be used in re-pricing @param idemandDepositAccount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	generatePricingContractContextDTO	This method would generate the pricing contract context dto using the demand deposit account as input which would then be used in re-pricing @param idemandDepositAccount @return	This method returns the filled RepriceFormDTO depending on the priceType in repriceFormDTO. The filling of the repriceFormDTO mainly concerns with determining whether the indexRate, marginRate and udmRate need to be retained by Pricing or not.  In case of CASA interest type or stage is OD or OverLine and if its in fixed period we should not re-price,not log rate change action OD/OverLine-variable period the re-price and log rate change action Credit/TOD-re-price and log rate change action AUF-if available then re-price and log rate change action TE-if available then re-price and log rate change action   @param accountId @param repriceDate @param repriceFormDTO @param rateChangeDetails
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	linkBundleBenefitsToArrangement	This method will stamp bundle variances to account. @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate	This method will stamp bundle variances to account. @param arrangementNumber @param partyId @param accountCurrency @param offerId @param bundleId @param bundleBenefitsApplicableFlag @param asOnDate
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	prepareBankingFactContextDTO	Preperation of BankingFactContextDTO for adding facts. @param adapterDTO @return	Method will prepare RuleContext.  {@link IBonusInterestFactsProvider} will be called to inquire all the fact details used in particular rule.That method will be implemented by consulting. @param ruleId @return context
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	prepareBankingFactContextDTO	Preperation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method will be used to calculate the interest benefit for the offset account case. The interest benefit would be calculated for the capitalization cycle. The balances for the account during the interest benefit period would be sent by the caller module. The interest would be calculated for the period based upon the balances sent.  @param interestBenefitRequest @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. The algorithm is as follows: &lt;ul&gt; If pricing ticket is not null &lt;li&gt;Indicating that the rate is already negotiated with the pricing module and same is required to be applied.&lt;/li&gt; &lt;li&gt;Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}&lt;/li&gt; &lt;li&gt;pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null &lt;li&gt;set other attributes in the input.&lt;/li&gt;&lt;/ul&gt; &lt;ul&gt; If pricing ticket is null &lt;li&gt;Indicating that there is no rate communication happened between pricing and the calling module.&lt;/li&gt; &lt;li&gt;Default rate is required to be applied.&lt;/li&gt; &lt;li&gt;Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()}&lt;/li&gt; &lt;li&gt;Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} to apply the variance and creating pricing plan&lt;/li&gt; &lt;/ul&gt; &lt;/ol&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	The method is responsible for creating backdated pricing plan in the pricing module. DDAPricingPlanAdapterDTO would include the account opening date as one of the attribute along with the process date. The algorithm is as follows: If pricing ticket is not null Indicating that the rate is already negotiated with the pricing module and same is required to be applied. Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null set other attributes in the input. If pricing ticket is null Indicating that there is no rate communication happened between pricing and the calling module. Default rate is required to be applied. Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()} Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])} to apply the variance and creating pricing plan   <b>Since:</b> R2.2  



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	<pre> &lt;ol&gt; Create BankingFactContextDTO from DepositsFeeRequestInfo attributes. &lt;li&gt;set bankingFactContextDTO.AccountCurrency = DepositsFeeRequestInfo.CurrencyCodes[0]&lt;/li&gt; &lt;li&gt;set bankingFactContextDTO.AccountId = DepositFeeRequestInfo.AccountId&lt;/li&gt; &lt;li&gt;set bankingFactContextDTO.PartyId = DepositFeeRequestInfo.PartyId&lt;/li&gt; &lt;li&gt;set bankingFactContextDTO.ProcessDate = DepositFeeRequestInfo.EvaluationDate&lt;/li&gt; &lt;li&gt;set bankingFactContextDTO.TransactionAmount = DepositFeeRequestInfo.TransactionAmount&lt;/li&gt; &lt;/ol&gt; @param feeRequestInfo @return </pre>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	<p>This method is used to inquire interest rate details for given {@link CASAInterestTypes}. Interest rate details contains following information.</p> <pre> &lt;ol&gt; &lt;li&gt;Base rate of interest- Attached Rate charts can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse specific rate chart to get effective base rate of interest. &lt;li&gt;Margin rate of interest- It can be either positive or negative. More than one rate charts can be attached to margin rate. Effective rate is addition of all interest rates. Each rate chart can be tiered or not tiered. For tiered rate chart instead of giving effective rate doing by processing, It just gives rate chart details. Consumers are expected to analyse each rate chart to get cumulative margin rate of interest. &lt;li&gt;UDM(user defined margin)- It is absolute value defined for the account in account level price benefit chart in pricing module. &lt;/ol&gt; This method fetches information from pricing module. This method performs following activities. &lt;ol&gt; &lt;li&gt;Prepares </pre>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				pricingContextDTO of type {@link PricingEventContextDTO};  Prepare FactContextDTO : Create instance of BankingFactContextDTO. <li>Call Pricing service: EventRateApplicationService#getRatesWithLenientComputation following parameter <ol> <li>pricingContextDTO <li>stage: input.interestRateType <li>evaluationDate : Current Process Date.(Get it from common service) </ol>   <li>Derive net interest rate: If base and margin rates are non tiered then net interest rate can be derived using following formula.   <code>Net rate = base rate + cumulative interest rate + UDM</code> @see DDAPricingApplicationService @see InterestRateDetailsDTO @param accountId @param interestRateType @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method creates banking fact context DTO with interest balance base facts and the corresponding clear amounts @param postingDate @param account @param accountCurrencyCode @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	During any financial transaction, this method will be called with the event id as input and if the event is eligible for reward, pricing is expected to return a reward due date. @param eventId @param accountId @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method is used to call pricing for persisting each stage for given account code with a given effective date, each stage will be stamped for account with stage start date, end date along with respective base rate and udm's.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method will be called to create and carry forward the existing UDM to new offer while offer swap @param adapterDTO @param eventRateVarianceDTOs @param forwardUDM @return ArrangementPlanLinkageKeyDTO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method will add a broker rebate margin for an od linkage which is linked to an existing account. @param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	prepareBankingFactContextDTO	Preparation of BankingFactContextDTO for adding facts. @param adapterDTO @return	This method is used to call pricing for persisting each stage for given account code with a given effective date, each stage will be stamped for account with stage start date, end date along with respective base rate and udm's.
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchExistingUDMs	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException	This method will be used to intimate removal of rebate when the overdraft facility is delinked from the account. It will check if there is any existing rebate UDM attached to the account for a given overdraft stage with fixed and variable rate type and negate the rebate margin for that account along with start date as expiry date +1 and call pricing to merge the new net rate.   @param accountId @param expiryDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchExistingUDMs	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException	In Amend OD account workflow. User can change various attributes of the account like Offer, OD amount, OD duration, Account Preferences etc. If product behaviour allows to reprice, we create new pricing recommendation based of new attributes. We create recommendation in following condition in amend workflow. <ol style="list-style-type: none"> <li>Offer swap is performed. <li>Increase / Decrease OD amount, provided bank policy permits repricing. </li> </li></ol> This method is used to create new recommendation. Following are steps involved. <ol style="list-style-type: none"> <li>Create new Price ticket. <li>Create facts based on new attributes. <li>Create new recommendation. </li> </li></li></ol>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchExistingUDMs	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException	This method will add a broker rebate margin for an od linkage which is linked to an existing account. @param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchExistingUDMs	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException	This method will return array of EventRateVarianceDTO if there is any existing UDM on Account. @param accountId @param sessionContext @return EventRateVarianceDTO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createPricingPlanAndForwardingExistingUDMs	This method will be called to create and carry forward the existing UDM to new offer while offer swap @param adapterDTO @param eventRateVarianceDTOs @param forwardUDM @return ArrangementPlanLinkageKeyDTO @throws FatalException	This method will be called to create and carry forward the existing UDM to new offer while offer swap. @param demandDepositAccount @param offerSwapDetails @param parentAccountId @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceOnEventStrictAtReco		This will provide the fees for event at the time of recommendation. This event is many times get called get called from the place where account is not opened yet. So at the time of showing the fees settlement payout gets called. So origination sets those fees and this service inquires those fees and show it on the screen. @param sessionContext @param accountId @param eventId @param amount @param pricingTicketId @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getPriceOnEventStrictAtReco		This will compute the price at the time of recommendation. This get called from settlement payout instruction from draft mode where account is not opened yet. @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	getFeeOnEventSimulation	<pre> This method is a wrapper over the Pricing module method to calculate the fees for a given event. The following is the algorithm for the method: &lt;ol&gt; &lt;li&gt; If Price ticketNumber is Null then, Call {@link com.ofss.fc.app.pricing.service.ticket.TicketService#startDialogue(SessionContext)} Method to acquire a {@link PricingTicketDTO}. &lt;/li&gt; &lt;li&gt; Get the Instance of an object by calling {@link getBankingFactContextDTO(DepositFeeRequestInfo: feeRequestInfo): BankingFactContextDTO} &lt;/li&gt; &lt;li&gt; Instantiate an object of {@link PricingEventContextDTO} &lt;/li&gt; &lt;li&gt; Set the </pre>	This method is used to fetch and calculate total fees for Account Closure Event. @param applicationContext @param accountId @param partyId @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>PricingTicketDTO} obtained earlier in this {@link PricingEventContextDTO}</p> <ul style="list-style-type: none"> <li>&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Instantiate {@link EventPriceApplicationService}</li> <li>&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; Call the method {@link EventPriceApplicationService#getPriceOnEvent()} which returns {@link PriceFeatureDTO} &amp;lt;/li&amp;gt;</li> <li>&amp;lt;/li&amp;gt; Copy the values from {@link PriceFeatureDTO} to an array of {@link PricingDetail} as follows: <ul style="list-style-type: none"> <li>&amp;lt;ul&amp;gt; &amp;lt;/li&amp;gt; feeAmount :</li> <li>PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</li> <li>prices.(MoneyBagDTO)netAmount</li> <li>&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt;</li> <li>feeDescription :</li> <li>PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</li> <li>prices.getDescription() [//TODO: getDescription() missing in ProcessedBasePriceDTO. Pricing team(Ramu) to take this up.]</li> <li>&amp;lt;/li&amp;gt; &amp;lt;/li&amp;gt; nsfTreatment :</li> <li>PriceFeatureDTO.List&amp;lt;ProcessedBasePriceDTO&amp;gt;</li> <li>prices.(ChargeInformationDTO)chargeInformation.nsfTreatment</li> <li>&amp;lt;/li&amp;gt; &amp;lt;/ul&amp;gt; &amp;lt;/li&amp;gt;</li> <li>&amp;lt;/li&amp;gt; Set the various attributes</li> </ul> </li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>of <a href="#">PricingDetail</a> with the attribute values present in <a href="#">PriceFeatureDTO</a> by iterating through the details in it.   The following are the parameters taken by the method along with their significance  <ul style="list-style-type: none"> <li>param feeRequestInfo : Information object containing the basic request details for fetching fees from the Pricing module. See <a href="#">DepositsFeeRequestInfo</a> for more details.</li> <li>return PricingDetail[] : The fee details for the given event. See <a href="#">PricingDetail</a> for more details.</li> </ul> <pre>@param feeRequestInfo @return @throws FatalException</pre> </p>	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	intimateReversalToPricingForReward	<p>9NT1487ST-1272: Method to intimate pricing to reverse counter for reward  @param pricingTicketNumber  @param transactionReferenceNumber  @param feeRequestInfo  @throws FatalException</p>	<p>This method will intimate pricing for reversal of reward for the original transaction  @param transactionType  @param accountTransaction  @param amountReversed</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	intimateReversalToPricingForReward	<p>9NT1487ST-1272: Method to intimate pricing to reverse counter for reward  @param pricingTicketNumber  @param transactionReferenceNumber  @param feeRequestInfo  @throws FatalException</p>	<p>This is utility method to intimate pricing based on rewardPricingTicket number.  For ex. In case of sweepout instruction fails, We need to intimate pricing for reversal of reward.  @param accountId  @param rewardPricingTicketnumber  @param originalTransactionReferenceNumber  @param eventId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	inquireDeferredRepriceActions	The method will be used to fetch DeferredRepriceActionInquiryResponse. @param arrangementId @param priceType @param triggeringFactCode @return	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	inquireDeferredRepriceActions	The method will be used to fetch DeferredRepriceActionInquiryResponse. @param arrangementId @param priceType @param triggeringFactCode @return	The method will update the reprice form DTO with the {@link RepriceProcessingAction} type. This will be set to {@link RepriceProcessingAction#DEFER} if the account is in fixed rate period. The {@link DDActionType#ODFixedRatePeriodEnd} action will be relogged with the extra attribute to indicate that pricing is to be called for Deferred repricing. Else {@link RepriceProcessingAction#ACCEPT} will be filled. <p> If the fact value change for which the repricing is proceeding already has an entry present in pricing, and the two fact values are different, {@link RepriceProcessingAction#CANCEL} will be sent. This will remove the deferred reprice action in pricing. </p> @param account @param repriceDate @param repriceFormDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	removeDeferredRepriceActions	The method will be used to remove deferred reprice actions. @param arrangementId @param priceType @param triggeringFactCode	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	removeDeferredRepriceActions	The method will be used to remove deferred reprice actions. @param arrangementId @param priceType @param triggeringFactCode	The method will update the reprice form DTO with the {@link RepriceProcessingAction} type. This will be set to {@link RepriceProcessingAction#DEFER} if the account is in fixed rate period. The {@link DDActionType#ODFixedRatePeriodEnd} action will be relogged with the extra attribute to indicate that pricing is to be called for Deferred repricing. Else {@link RepriceProcessingAction#ACCEPT} will be filled. <p> If the fact value change for which the repricing is proceeding already has an entry present in pricing, and the two fact values are different, {@link RepriceProcessingAction#CANCEL} will be sent. This will remove the deferred reprice action in pricing. </p> @param account @param repriceDate @param repriceFormDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	processDeferredRepriceActions	The method will process deferred reprice actions. @param arrangementId @param domainCategoryCode @param repriceDate	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	processDeferredRepriceActions	The method will process deferred reprice actions. @param arrangementId @param domainCategoryCode @param repriceDate	The method will be used to execute {@link DDActionType#ODFixedRatePeriodEnd} action. It does the following: <ol> In fixed period expire process, The account should move ahead with new rates from next day of due date. <li>If due date is today: <p> account should move ahead with new rates from next day. In this case, system logs {@link DDActionType#BODRateChange} because it can take care of scenarios for (due date + 1) is working or holiday. </p> </li> <li>If due date is back date: <p> account should move ahead with new rates from the next date of date of expire. So system executes {@link DDActionType#BackdatedTransaction}. So interest fetches new rates from pricing from that day. </p> </li> </ol> @param applicationContext

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param actionKey @param dueDate
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchInterestRateDetailsForBackdated	This method will be used in case of backdated account opening where it would call pricing internally and do the following ContractApplicationService.computeFeesAndPersistRatesConsideringRepriceEvents(): This method will reprocess all reprice actions like index/rate chart/ppc/fees changes from backdated value date till current posting date. While reprocessing each of this reprice event, pricing will call back following API of CASA: DDAPricingAccountAdapter.getRepricingConfiguration() with RepriceFormDTO.RepricePurpose=BACKDATED_REPRICE. CASA should fill RepriceFormDTO providing details of reprice. This method will persist rates from back value date till infinity for given stages and return Periodic Fee computation dates. Module should log periodic fee computation actions for these dates. This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when margin is included. User Descretionary margin (UDM) is given for a period only. It has start	This method is used to update rate of interest. This method get latest interest rate from pricing module and populate it into account. This can be used at the time of account opening and re-pricing to update the interest rate in account. This is used to update only non-tired interest rate. @param sessionContext @param accountId @return {@link Boolean}

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is gettingt expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.execute Action which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAPricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAPricingAdapter	persistRatesForBackdate	This method is used to call pricing for persisting each stage for given account code with a given effective date,each stage will be stamped for account with stage start date,end date along with respective base rate and udm's. In case of backdated account opening for periodic fees, a PeriodicEventTrackerDTO will be returned from pricing which demand deposit needs to use and log a periodic fee action.	This method will be used in case of backdated account opening where it would call pricing internally and do the following ContractApplicationService.computeFeesAndPersistRatesConsideringRepriceEvents(): This method will reprocess all reprice actions like index/rate chart/ppc/fees changes from backdated value date till current posting date. While reprocessing each of this reprice event, pricing will call back following API of CASA: DDAPricingAccountAdapter.getRepricingConfiguration() with RepriceFormDTO.RepricePurpose=BACKDATED_REPRICE. CASA should fill RepriceFormDTO providing details of reprice. This method will persist rates from back value date till infinity for given stages and return Periodic Fee computation dates. Module should log periodic fee computation actions for these dates. This method will fetch the interest rate details that will be required for updating rate in Demand deposit account. The rate details are not be returned when magin is included. User Descretionary margin (UDM) is given for a period only. It has start and end date indicating the period when the UDM is effective. If UDM is applicable from a future date after account opening then a future dated rate change action should be logged. It is not require to log rate change action on start date if UDM start date is already passed on account opening date. If UDM is gettingt expired on a future date then also a future dated rate change action needs to be logged. Future dated UDM action for expiry date will be logged from next date of UDM expiry date. If the expiry date is not present indicating no expiry of the UDM and hence no action will be logged. UDM can be fixed or variable hence logging

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>action in both the cases. Rate change action is required to be logged at the time of account opening only. When it is a call from account opening service then it will have pricingPlanId as an input. If pricing plan id is not passed and value is null then this is a call from RateChangeActionExecutor.executeAction which does not pass pricing plan id as input. @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistAndLogRateChangeBackdated	<p>This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage, od force end, od fixed period, od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws FatalException</p>	<p>This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows. &lt;ul&gt; &lt;li&gt;Fetch latest limits from LCM. Stamp it on account. The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.&lt;/li&gt; &lt;li&gt;Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date, od end date on account. These are parameters which are common for linkage on current date and for back date.&lt;/li&gt; &lt;li&gt;Now it finds out the list of {@link BalanceHistoryItem} and updates them with sanctioned OD limit amount and available amount on that date by using {@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)} &lt;li&gt;After that it fetches the sanctioned limit from facility by using {@link IDDLCAAdapter#fetchEffectiveLimitOfFacility(String)}. The sanctioned limit can not be directly used for</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				align because there can be other accounts connected to same facility. So it fetches the effective limit of facility by using {@link IDDLCAAdapter#fetchBalanceDetails(String)}. Now this limit is used for reapportion. </li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not. </li> <li>It intimates LCM for linkage of facility and account. So that LCM can realign their account with proper facilities </li> <li>It calls reapportion for aligning the funds with CASA and LCM. </li> <li>It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging. </li> <li>Now it intimates the NPA, Collection module about the linkages. </li> </ul> @param demandDepositAccount @param accountLinkage @param isODRateFixed
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistAndLogRateChangeBackdated	This method will be used for persisting the rate changes intimation to pricing along with logging a rate change action for UDM and margin rate changes for stages depending on the stage input by the caller. This method will be called in case of OD start linkage, od force end, od fixed period, od fixed expiry In case of other stage than od it will log persist rate for that stage and log udm and margin rate @param accountId @param partyId @param accountCurrencyCode @return {@link AccountBalanceRateDTO} @throws	This method works in this flow. <ol> <li>If borrowing entity id in DemandDepositAccount is null then in that case it creates Borrowing entity id for given account and stamps it on account. It ensures account can use limits and collatrels with that borrowing entity id. </li> <li>If start date of borrowing entity is more than current date due to borrowing entity created after cutoff in that case start date is replaced with current date. </li> <li>Then it sets limit on account. LCM has to ensure to call CASA before aligning their funds to ensure proper limit intimation. </li> <li>If temporary excess is null on account then it creates them, then it stamps the sanctioned limit, available limit. </li> <li>If start date of facility is not equal to current date and it is in batch mode then it is case of backdated TE linkage due to

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			FatalException	<p>holiday processing. In this case it asks pricing for backdated linkage of stage. <a href="#">{@link IDDA PricingAdapter#persistAndLogRateChangeBackdated(String, Date, CASAInterestTypes, boolean)}</a></p> <p><li>If start date of facility equals to current then it asks pricing for current dated stage creation. <a href="#">{@link IDDA PricingAdapter#persistAndLogRateChange(String, Date, CASAInterestTypes, boolean)}</a></li> <li>It first now aligns current state of account by using <a href="#">{@link DemandDepositAccount#reapportionFunds(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)}</a> for non batched linkage and <a href="#">{@link DemandDepositAccount#reapportionFundsForAmountChange(String, FinancialEventType, MessageType, NSFTreatment, TransactionDates, TransactionAttributes, boolean)}</a> for batched linkages.</li> <li>Now for history items in range it aligns them by using <a href="#">{@link this#mergeSanctionAmountForHistoryItemsInRange(Date, DemandDepositAccount, Money, ApplicationContext, TransactionAttributes, TransactionDates, com.ofss.fc.enumeration.lcm.limits.FacilityType)}</a></li> <li>Lastly it calls <a href="#">{@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)}</a> for handling action part of linkage.</li> <p><code>@param accountId @param facilityId @param limitAmount @param startDate @param endDate</code></p> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchCapitalizedFeesForArrangementNumber	On OD Account opening, origination marks some of the fees as capitalised. Capitalised means fee amount will be added in OD amount. At the time of disbursement , DDA module fetch pending capitalised fees from pricing and charge it. @param accountId	This method handles differed fee handling for given casa account.It fetches the differed fees from pricing and then charges it using fee helper. @param accountId @param debitInstruction @param applicationContext @param feeHelper @param ddaPricingAdapter @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchCapitalizedFeesForArrangementNumber	On OD Account opening, origination marks some of the fees as capitalised. Capitalised means fee amount will be added in OD amount. At the time of disbursement , DDA module fetch pending capitalised fees from pricing and charge it. @param accountId	On OD Account opening, origination marks some of the fees as capitalised. Capitalised means fee amount will be added in OD amount. At the time of disbursement , DDA module fetch pending capitalised fees from pricing and charge it. @param accountId
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchNetRate	This method would return the net rate for given date and stage. @param arrangementNumber @param effectiveDate @param stage @param dateToFilterOn @return	This method will actually log the alert based on tiered or non-tiered rate changes have taken place. @param actionKey @param effectiveDate @param interestType @param eventCode
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchNetRate	This method would return the net rate for given date and stage. @param arrangementNumber @param effectiveDate @param stage @param dateToFilterOn @return	This method is used to @param accountId @return
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchNetRate	This method would return the net rate for given date and stage. @param arrangementNumber @param effectiveDate @param stage @param dateToFilterOn @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateArrangementIdToAccountIdForFees	<p>&lt;ol&gt; &lt;li&gt;Module inquire fees based on fee ticket number and Recommendation ticket number.This time account is not generated. Pricing gives fees based on recommendation created. &lt;li&gt;At the time of charging/account opening ,We intimate pricing to update account number which is charged. &lt;/ol&gt; @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException</p>	<p>This method would mark fees as capitalized in pricing. @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateArrangementIdToAccountIdForFees	<p>&lt;ol&gt; &lt;li&gt;Module inquire fees based on fee ticket number and Recommendation ticket number.This time account is not generated. Pricing gives fees based on recommendation created. &lt;li&gt;At the time of charging/account opening ,We intimate pricing to update account number which is charged. &lt;/ol&gt; @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException</p>	<p>This method would mark fees as deferred in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	updateArrangementIdToAccountIdForFees	<p>&lt;ol&gt; &lt;li&gt;Module inquire fees based on fee ticket number and Recommendation ticket number. This time account is not generated. Pricing gives fees based on recommendation created. &lt;li&gt;At the time of charging/account opening ,We intimate pricing to update account number which is charged. &lt;/ol&gt; @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException</p>	<p>&lt;ol&gt; &lt;li&gt;Module inquire fees based on fee ticket number and Recommendation ticket number. This time account is not generated. Pricing gives fees based on recommendation created. &lt;li&gt;At the time of charging/account opening ,We intimate pricing to update account number which is charged. &lt;/ol&gt; @param sessionContext @param pricingTicketNumber @param accountId @throws FatalException</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	markFeeTransactionAsComplete	<p>Given an ticket Number this method will load all the Fees for that ticketNumber and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException</p>	<p>This method would mark fees as capitalized in pricing. @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	markFeeTransactionAsComplete	Given an ticket Number this method will load all the Fees for that ticketNumber and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException	This method would mark fees as deferred in pricing @param sessionContext @param applicationContext @param outlayFundsDTO @param accountId @param currencyCodes @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	markFeeTransactionAsComplete	Given an ticket Number this method will load all the Fees for that ticketNumber and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param	This method is responsible for update pricing that a transaction pertaining to fees is completed.   This is required for pricing to distinguish the transaction from a simulation transaction to an actual transaction   @param pricingTicketNumber @param eventCodes

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			ticketNumber @param eventCodes @param chargeActionType @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	markFeeTransactionAs Complete	Given an ticket Number this method will load all the Fees for that ticketNumber and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException	Given an ticket Number this method will load all the Fees for that ticketNumber and then update the flag 'isTransactionComplete' to true which mean that the fees which were simulated from the screen are actually charged to the account. Based on the charge action type passed the charge details for transaction will be updated. Note: Transactions cannot be Charged Collected using this method, it throws an INVALID_CHARGE_ACTION_TYPE in this case. @param sessionContext @param ticketNumber @param eventCodes @param chargeActionType @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchFeesOnEventForRecommendation	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number. Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number. Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchFeesOnEventForRecommendation	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number. Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException	In Amend OD account workflow. User can change various attributes of the account like Offer, OD amount , OD duration, Account Preferences etc. If product behaviour allows to reprice, we create new pricing recommendation based of new attributes. We create recommendation in following condition in amend workflow. <ol> <li>Offer swap is performed. <li>Increase / Decrease OD amount, provided bank policy permits repricing. </ol>   This method is used to create new recommendation. Following are steps involved. <ol> <li>Create new Price ticket. <li>Create facts based on new attributes. <li>Create new recommendation. </ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchFeesOnEventForRecommendation	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number.Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException	In Manage facility, Using existing facility , New OD account is created.This method is used to create new recommendation while opening account.Following are steps involved. <ol> <li>Create new Price ticket. <li>Create facts based on new attributes. <li>Create new recommendation. </ol> @param sessionContext @param pricingRequestDTO @param factMap @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	fetchFeesOnEventForRecommendation	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing modules provides fee based on recommendation ticket number.Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException	When account is not opened, This method gives price of servicing events using recommendation ticket.Pricing recommendations are created in origination workflow.   Following are the steps performed in this method. <ol> <li>Prepare fee request -Prepare Facts using {@link ODAccountConfigurationParameterDTO}. <li>Call pricing service for list of events specified in {@link DDAccountConfigurationFeeRequestDTO#getEvents()}. <li>Prepare and store response-> <ol> <li>At the time of servicing fee inquiry,We store details using pricing ticket number. <li>At the time of account opening , We mark stored fees as deferred or capitalised using stored details. <li>At the time of OD settlement,capitalised fees gets charged. <li>If OD settlement is not present then {@link #levyFeesOnAccountConfiguration(SessionContext, DDAccountConfigurationFeeChargingRequestDTO)} is called. </ol> </ol>   Response contains capitalised and non capitalised fee. <ol>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<ul style="list-style-type: none"> <li>&lt;li&gt;&lt;b&gt;Capitalised Fee&lt;/b&gt;: It gets added in OD limit.If customer wants OD limit of 1000 AUD and 50 AUD are servicing fees.So limit will be 1050 AUD. 50 Aud will be deducted in OD settlement execution.</li> <li>&lt;li&gt;&lt;b&gt;Non-Capitalised/Deffered Fee&lt;/b&gt; : Does not get added in OD limit,If customer wants OD limit of 1000 AUD and 50 AUD is servicing fee. So limit will remain same i.e 1000 AUD. But 50 AUD will be deducted in OD settlement execution. &lt;/ol&gt; @param sessionContext @param accountServicingFeeRequestDTO @return @throws FatalException</li> </ul>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchFeesOnEventForRecommendation	Method used to fetch fees based on recommendation pricing ticket number.   When OD account is not opened and servicing fees are required. Pricing module provides fee based on recommendation ticket number.Following steps are performed . <ol> <li>Create New fee ticker number. <li>Create DDA pricing event context. <li>Call Pricing service with old recommendation and new fee ticket number. </ol> @param feeRequestInfo @return @throws FatalException	Method used to prepare fee request when we want to fetch fees from recommendation ticket number. @param sessionContext @param demandDepositAccount @param accountServicingFeeRequestDTO @param listOfEvents @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	basicIntimateReversalToPricing		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createPricingRecommendationOnAmendOD	This method create new pricing recommendation from amend od configuration.   We create recommendation in following condition in amend workflow. <ol><li>Offer swap is performed.</li><li>Increase / Decrease OD amount, provided bank policy permits repricing. </ol> @param sessionContext @param pricingRequestDTO @param currencyCode @param factMap @return @throws FatalException	In Amend OD account workflow.User can change various attributes of the account like Offer,OD amount , OD duration,Account Preferences etc. If product behaviour allows to reprice, we create new pricing recommendation based of new attributes. We create recommendation in following condition in amend workflow. <ol><li>Offer swap is performed.</li><li>Increase / Decrease OD amount, provided bank policy permits repricing. </ol>  This method is used to create new recommendation.Following are steps involved. <ol><li>Create new Price ticket.</li><li>Create facts based on new attributes. </li><li>Create new recommendation. </ol>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createPricingRecommendationOnAmendOD	This method create new pricing recommendation from amend od configuration.   We create recommendation in following condition in amend workflow. <ol><li>Offer swap is performed.</li><li>Increase / Decrease OD amount, provided bank policy permits repricing. </ol> @param sessionContext @param pricingRequestDTO @param currencyCode @param factMap @return @throws FatalException	In Manage facility, Using existing facility , New OD account is created.This method is used to create new recommendation while opening account.Following are steps involved. <ol><li>Create new Price ticket.</li><li>Create facts based on new attributes. </li><li>Create new recommendation. </ol> @param sessionContext @param pricingRequestDTO @param factMap @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	createPricingRecommendationOnAmendOD	This method create new pricing recommendation from amend od configuration.   We create recommendation in following condition in amend workflow. <ol><li>Offer swap is performed.</li><li>Increase / Decrease OD amount, provided bank policy permits repricing. </ol> @param sessionContext @param pricingRequestDTO @param currencyCode @param factMap @return @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateRebateRemovalToPricing	This method will be used to intimate removal of rebate when the overdraft facility is delinked from the account.It will check if there is any rebate UDM attached the account and negate the rebate margin for that account along with start date as expiry date +1 and call pricing to merge the new net rate.   @param accountId @param expiryDate	@param demandDepositAccount
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	intimateRebateRemovalToPricing	This method will be used to intimate removal of rebate when the overdraft facility is delinked from the account.It will check if there is any rebate UDM attached the account and negate the rebate margin for that account along with start date as expiry date +1 and call pricing to merge the new net rate.   @param accountId @param expiryDate	This method will be used to intimate removal of rebate when the overdraft facility is delinked from the account.It will check if there is any existing rebate UDM attached to the account for a given overdraft stage with fixed and variable rate type and negate the rebate margin for that account along with start date as expiry date +1 and call pricing to merge the new net rate.   @param accountId @param expiryDate

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	addUDMtoPPC	This method will add a broker rebate margin for an od linkage which is linked to an existing account. @param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException	This method will perform the job of linking given od facility to given demandDepositAccount on current posting date. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCAAdapter#fetchEffectiveLimitOffFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCAAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.</li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper facilities</li> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls {@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)} for handling action part of logging.</li> <li>Now it intimates the NPA,Collection module about the linkages.</li> </ul> @param demandDepositAccount @param accountLinkage

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param isODRateFixed
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	addUDMtoPPC	This method will add a broker rebate margin for an od linkage which is linked to an existing account. @param sessionContext @param productGroupLinkageType @param effectiveDate @param accountId @param udmValue @throws FatalException	This method will perform the job of linking given od facility back dated to given demandDepositAccount. The steps involved in the processing are as follows. <ul> <li>Fetch latest limits from LCM.Stamp it on account.The method which used for this is {@link DemandDepositUtil#setLimits(DemandDepositAccount, TransactionDates)} here transaction dates object contains the current posting date for all the dates inside transaction dates.</li> <li>Now it processes and stamps the common parameters such as borrowing entity id, borrowing entity start date , od start date,od end date on account.These are parameters which are common for linkage on current date and for back date.</li> <li>Now it finds out the list of {@link BalanceHistoryItem} and updates them with sanctioned OD limit amount and available amount on that date by using {@link ODLimitLinkageExecutor#handleBalanceHistoryItemsInRange(UtilizationDetailsResponse, AccountFacilityLinkage, Money)} <li>After that it fetches the sanctioned limit from facility by using {@link IDDLCMAdapter#fetchEffectiveLimitOfFacility(String)}.The sanctioned limit can not be directly used for align because there can be other accounts connected to same facility.So it fetches the effective limit of facility by using {@link IDDLCMAdapter#fetchBalanceDetails(String)}.Now this limit is used for reapportion.</li> <li>Now it calls pricing for logging rate change for OD stage by sending details of fixed rate or not.</li> <li>It intimates LCM for linkage of facility and account.So that LCM can realign their account with proper

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>facilities</p> <ul style="list-style-type: none"> <li>It calls reapportion for aligning the funds with CASA and LCM.</li> <li>It calls <code>{@link DemandDepositUtil#handleBalanceChangeAction(DemandDepositAccount, Date, TransactionAttributes)}</code> for handling action part of logging.</li> <li>Now it intimates the NPA,Collection module about the linkages.</li> </ul> <p>@param demandDepositAccount @param accountLinkage @param isODRateFixed</p>
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	mergeRebateVarianceToRecommendationForAmendOD	This method is used to update rebate margin for Overdraft stage. Method removes previous rebate margin applied and adds current margin given in amend configuration. @param sessionContext @param amendConfigurationDTO @throws FatalException	This method is used to update rebate margin for Overdraft stage. Method removes previous rebate margin applied and adds current margin given in amend configuration. @param sessionContext @param amendConfigurationDTO
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	mergeRebateVarianceToRecommendationForAmendOD	This method is used to update rebate margin for Overdraft stage. Method removes previous rebate margin applied and adds current margin given in amend configuration. @param sessionContext @param amendConfigurationDTO @throws FatalException	
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRatesForArrangementOnGivenAndPreviousDate	This method fetches applicable Net rates for given arrangementNumber on asOnDate and its previous date @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException	This method fetches applicable Net rates for given arrangementNumber on asOnDate and its previous date @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRatesForArrangementOnGivenAndPreviousDate	This method fetches applicable Net rates for given arrangementNumber on asOnDate and its previous date @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException	Log entry into the non financial statement for the rate change event.  It would call pricing to get the current date rate and the immediate previous date rate.   Using the response , the narration would be constructed and logged in the non financial item. @param arrangementNumber @param asOnDate @param eventCode @param serviceId @param valueDate TODO @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	fetchRatesForArrangementOnGivenAndPreviousDate	This method fetches applicable Net rates for given arrangementNumber on asOnDate and its previous date @param sessionContext @param arrangementNumber @param asOnDate @return @throws FatalException	This method is to fetch the current date rates and immediate previous date rate. @param arrangementNumber @param asOnDate @return @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA PricingAdapter	restructure FeesAndRatesConsideringRepriceEvents	This method is used to restructure rates and fees. As of now it is called in following situation. <ol> <li>If repricing is performed after domain fact change. </ol> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	restructure FeesAndRatesConsideringRepriceEvents	This method is used to restructure rates and fees. As of now it is called in following situation. <ol style="list-style-type: none"> <li>If repricing is performed after domain fact change. </li> </ol> @param applicationContext @param arrangementNumber @param repriceFormDTO @param restructureDate @param currencyCode @param fixedPeriod @param considerVarianceChanges @param computeFeesForLapsedPeriod @param prorateWithAlreadyChargedFees @return	If any fact which is used in pricing details, pricing module asks whether to reprice or not. It does following activities. <ol style="list-style-type: none"> <li>Derives the event for fact change. <a href="#">{@link #fetchRepriceEventForFactChange(RepriceFormDTO)}</a> . <li>Check reprice event configuration at bank policy. <a href="#">{@link #isAccountToBeRepricedOnFactChange(IDemandDepositAccount, String, Date)}</a> <li>Call <a href="#">{@link #fillRepriceFormDTO(String, Date, RepriceFormDTO)}</a> to include reprice details <li>Call pricing method to restructure fees provided if repricing response is yes. </li> </li></li></li></ol> @param arrangementNumber, DDA account number @param repriceFormDTO, fact change reprice details. @param repriceDate, reprice date, it can be backdated.
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	createPricingRecommendationOnManageFacility	In Manage facility, Using existing facility , New OD account is created.This method is used to create new recommendation while opening account.Following are steps involved. <ol style="list-style-type: none"> <li>Create new Price ticket. </li> <li>Create facts based on new attributes. </li> <li>Create new recommendation. </li> </ol> @param sessionContext @param pricingRequestDTO @param factMap @return @throws FatalException	This method is used to create pricing details for overdraft account which is to be created. <ol style="list-style-type: none"> <li>In Manage facility, We open OD account under existing facility. </li> <li>Before opening account, Pricing recommendation is shown using overdraft information provided. </li> <li>This method is used to create/fetch pricing recommendation details. </li> <li>OD information will be used to derive pricing rates and fees. </li> <li>Recommendation ticket number is <a href="#">{@link NetInterestRateResponse#getPricingTicketNumber()}</a> is used to track pricing details in future. </li> <li>On Account opening <a href="#">{@link #openODAccount(SessionContext, ODAccountConfigurationDTO)}</a> , Recommendation ticket number is used to create pricing plan. </li> </ol> @param sessionContext, SessionContext ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @param odAccountPricingInputDTO,

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				Pricing input must contain OD configuration details. @return {@link NetInterestRateResponse}, contains pricing ticket number to track pricing details. @throws FatalException
IDDA Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDA Pricing Adapter	persistInterestRateDetailsForAmendmentOD	This method persist rates in amend OD flow. Fixed period rates of OD also changed. @param accountId @param partyId @param accountCurrencyCode @param evaluationDate @param arrangementPlanLinkageKeyDTO @return @throws FatalException	<p>&lt;ol&gt;The method is responsible for creating pricing plan in the pricing module. It works as follows:</p> <ul style="list-style-type: none"> <li>Create AdpaterFactory and get an instance of DDAPricingAdapter</li> <li>Call {@link IDDA Pricing Adapter#createPricingPlanAndEndDialogue(priceTicketNumber: String, accountNumber: String)} to create the pricing plan</li> </ul> <p>Creating PricingPlan Works as follows::</p> <ul style="list-style-type: none"> <li>If pricing ticket is not null <ul style="list-style-type: none"> <li>indicating that the rate is already negotiated with the pricing module and same is required to be applied.</li> <li>Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</li> <li>pass EventRateVarianceDTO[], EventPriceVarianceDTO[] both as null</li> <li>set other attributes in the input.</li> </ul> </li> <li>If pricing ticket is null <ul style="list-style-type: none"> <li>Indicating that there is no rate communication happened between pricing and the calling module.</li> <li>Default rate is required to be applied.</li> <li>Find the recommended rate by calling {@link ContractApplicationService#getPriceOffering()}</li> <li>Call {@link ContractApplicationService.linkVariancesAndEndDialogue(SessionContext, PricingContractContextDTO, String, Date, EventRateVarianceDTO[], EventPriceVarianceDTO[])}</li> </ul> </li> </ul> <p>to apply the variance and creating pricing plan</p> <p>@param priceTicketNumber @param demandDepositAccount</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				@param postingDate @param amendConfigurationDTO @param parentAccountId @throws FatalException
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	addAgentArrangement Linkage	Used to call {@link AgentArrangementLinkageApplicationService#addAgentArrangementLinkage} @param agentArrangementLinkageDTO	
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	addAgentArrangement Linkage	Used to call {@link AgentArrangementLinkageApplicationService#addAgentArrangementLinkage} @param agentArrangementLinkageDTO	Method will be called to persist agent arrangement details. @param agentArrangementLinkageDTO
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	fetchAgentArrangement Linkage	Used to call {@link AgentArrangementLinkageApplicationService#fetchAgentArrangementLinkage} @param accountId @return	
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	fetchAgentArrangement Linkage	Used to call {@link AgentArrangementLinkageApplicationService#fetchAgentArrangementLinkage} @param accountId @return	This method will get the rebate margin value to be used when an OD linkage is done to an existing account which has an broker/introducer attached to the account. @param accountFacilityLinkageDTO @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	fetchAgentArrangementLinkage	Used to call {@link AgentArrangementLinkageApplicationService#fetchAgentArrangementLinkage} @param accountId @return	This represents the fetchAccountFacilityLinkage method of the entity. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,AccountFacilityLinkageDTO AccountFacilityLinkageDTO. It contains following details <ol> <li>AccountFacilityLinkageKeyDTO:Key for entity AccountFacilityLinkage, contains account Id and Facility Id, using this information we can inquire about a facility linked to a account.i.e its start date , end date , fixed rate duration if applied. <li>fixedRatePeriod:Fixed rate period defines the start date end date for the period through which the account is charged a fixed rate.fixed rate period start date should always be equal to linkage period start date. <li>linkagePeriod:Linkage Period defines the Start Date and End Date for the facility linked to the account </ol> @return {@link com.ofss.fc.app.dda.dto.account.balance.AccountFacilityLinkageInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IDDAAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAAgentAdapter	fetchAgentArrangementLinkage	Used to call {@link AgentArrangementLinkageApplicationService#fetchAgentArrangementLinkage} @param accountId @return	Method will be called to fetch agent agreement linkage details. @param agentArrangementLinkageDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAgentAdapter	delinkAccountCommission	This method is used to copy the provided arrangement ID record to history and then delete the record from main agent table. This will be called mainly in case when an account gets delinked and therefore in agent DB the corresponding broker has to be dropped for that account. @param accountId	This method is used to copy the provided arrangement ID record to history and then delete the record from main agent table. This will be called mainly in case when an account gets delinked and therefore in agent DB the corresponding broker has to be dropped for that account. @param accountId
IDDAgent Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAgentAdapter	delinkAccountCommission	This method is used to copy the provided arrangement ID record to history and then delete the record from main agent table. This will be called mainly in case when an account gets delinked and therefore in agent DB the corresponding broker has to be dropped for that account. @param accountId	@param demandDepositAccount
IPCPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPricingAdapter	reverseFees	This method calls the casa fee-management service for reversing fees. @param accountID @param eventId @param messageType @param transactionType @param narration @param transactionDescription @throws FatalException	
IPCPricingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPricingAdapter	reverseFees	This method calls the casa fee-management service for reversing fees. @param accountID @param eventId @param messageType @param transactionType @param narration @param transactionDescription @throws FatalException	This method does the reversal of fees while reversing the transaction. @param accountId @param eventId @param narration @param transactionDescription @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCPrisingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPrisingAdapter	reverseFees	This method calls the casa fee-management service for reversing fees. @param accountID @param eventId @param messageType @param transactionType @param narration @param transactionDescription @throws FatalException	@see com.ofss.fc.app.pc.billpay.service.biller.IBillPaymentService#reverseBillPayment(com.ofss.fc.app.context.SessionContext, com.ofss.fc.enumeration.dda.MessageType)
IPCPrisingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPrisingAdapter	reverseFees	This method calls the casa fee-management service for reversing fees. @param accountID @param eventId @param messageType @param transactionType @param narration @param transactionDescription @throws FatalException	@param sessionContext @param bankDraft @param pricingAdapter @param eventId @param transactionType @throws FatalException
IPCPrisingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPrisingAdapter	reverseFees	This method calls the casa fee-management service for reversing fees. @param accountID @param eventId @param messageType @param transactionType @param narration @param transactionDescription @throws FatalException	<p>This method will reverse the fee charged for a transaction. </p> @param accountId @param accountModule @param eventId @param serviceId @param narrative @param atmNarration @param txnLiteral @param txnDescription
IPCPrisingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPrisingAdapter	getDialogueTicketOnEvent	This method simply obtains a pricing ticket for further dialogue. @return @throws FatalException	
IPCPrisingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPrisingAdapter	getPriceOnEvent	@param pricingContextDTO @param arrangementNumber @param eventCode @param asOnDate @param currencyCodes @return @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPC Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPC Pricing Adapter	getPriceOnEvent	@param pricingContextDTO @param arrangementNumber @param eventCode @param asOnDate @param currencyCodes @return @throws FatalException	@param transactionLog @param messageLogItem @throws FatalException
IPC Pricing Adapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPC Pricing Adapter	getWrappedPriceOnEvent	@param pricingContextDTO @param arrangementNumber @param eventCode @param asOnDate @param currencyCodes @return @throws FatalException	
IROSOProcessForApprovalCallingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\collection\IROSOProcessForApprovalCallingAdapter	callApproval	@param ROSOProcessForApprovalRequestDTO @return TransactionStatus @throws FatalException	@param ROSOProcessForApprovalRequestDTO @return TransactionStatus @throws com.ofss.fc.infra.exception.FatalException
IPartyKYCCheckAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyKYCCheckAdapter	performOnlineKYCCheck		This method calls adapter for executing online KYC. @param kycHistoryDTO @return KYCHistoryDTO
IPartyKYCCheckAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyKYCCheckAdapter	performOnlineKYCCheck		@return Object
IPartyKYCCheckAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyKYCCheckAdapter	performOnlineKYCCheck		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	Checks whether the Gaap Code matches in the inputs @param iAccountingEntryClass
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	This is the main validate method @param productEventLinkage
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	Checks whether the Gaap Code matches in the inputs @param iAccountingEntryTemplate
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	this is the main validate method @param productEventLinkage
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateGaapCode	This method validates the GAAP Code sent with the Accounting Services @param gaapCode String @return isValidGaap boolean	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateLedgerCode	This method validates the Account head with the ledger codes maintained in the accounting services. @param ledgerCode @param accountRoleType @param isCustomerLedger @return	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateLedgerCode	This method validates the Account head with the ledger codes maintained in the accounting services. @param ledgerCode @param accountRoleType @param isCustomerLedger @return	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	validateLedgerCodeforProductGroup	@param ledgerCode @param accountRoleType @param isCustomerLedger @param isMappedModuleSuppliedRole @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	checkAccountingSetup	<p>&lt;p&gt;This method will check that product group event linkage is defined for the input product group code. If not found, this service will return false, else it will check that template is defined for all input event list. If template is missing for any, it will return false. If all events are found, it will identify all account roles in the template list that start with the input pd name. For each of the identified account roles, role to ledger mapping must exist, else this service will return false. &lt;/p&gt;</p> <p>@param productGroupCode  @param eventCodeSet @param pdCodeSet @return</p>	
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	checkAccountingSetup	<p>&lt;p&gt;This method will check that product group event linkage is defined for the input product group code. If not found, this service will return false, else it will check that template is defined for all input event list. If template is missing for any, it will return false. If all events are found, it will identify all account roles in the template list that start with the input pd name. For each of the identified account roles, role to ledger mapping must exist, else this service will return false. &lt;/p&gt;</p> <p>@param productGroupCode  @param eventCodeSet @param pdCodeSet @return</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	checkAccountingSetup	<p>&lt;p&gt;This method will check that product group event linkage is defined for the input product group code. If not found, this service will return false, else it will check that template is defined for all input event list. If template is missing for any, it will return false. If all events are found, it will identify all account roles in the template list that start with the input pd name. For each of the identified account roles, role to ledger mapping must exist, else this service will return false. &lt;/p&gt;           @param productGroupCode            @param eventCodeSet @param pdCodeSet @return</p>	<p>&lt;p&gt;This method will do the following for the input product group code: &lt;ol&gt; &lt;li&gt;Validate Price Policy Chart for Interest&lt;/li&gt; If fail, add exception - Price Policy Chart (Interest) not defined for the selected Product Group. &lt;br/&gt; &lt;li&gt;Check if fee ppc exists and if it does perform the below, else update product group as validated. &lt;ol&gt; &lt;li&gt;Validate Product Group Role Mapping has been performed for the selected Product Group and PPC (Fees)&lt;/li&gt; If fail, add exception - Product Group Role Mapping not performed for the selected Product Group. &lt;br/&gt; &lt;/ol&gt; &lt;li&gt;Validate For Product Group which is Asset, validate whether a PPC (Regular Interest) is defined for the ΓÇÿDomain CategoryΓÇÖ as ΓÇ£OriginationΓÇ?&lt;/li&gt; If fail, add exception - Price Policy Chart (Regular Interest) not defined for Origination domain category for the selected Product Group. &lt;/ol&gt; &lt;/p&gt; @param productGroup</p>
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	isAccountingTemplateDefined	<p>&lt;p&gt;This method will check if accounting template is defined for the input product code. &lt;/p&gt;           @param productCode @return</p>	<p>&lt;p&gt;This method performs the following validations: &lt;ol&gt; &lt;li&gt;Whether an Interest Rate Definition (IRD) is defined for the input product.&lt;/li&gt; If fail, add exception - Interest Rate Definition (IRD) not defined for the selected Product.&lt;br/&gt; &lt;li&gt;Whether an Accounting Template is defined for the input product.&lt;/li&gt; If fail, add exception - Accounting Template not defined for the selected Product. &lt;br/&gt; &lt;ol&gt; &lt;li&gt;If templates exist, system will check if there exists a corresponding GL mapping for each of the Account Roles linked in the maintenance for the input product code.&lt;/li&gt; If fail, add exception - Account Role Mapping not defined for the selected Product. &lt;/ol&gt; &lt;/ol&gt; &lt;/p&gt; @param product</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPMUAccountingAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pm\IPMUAccountingAdapter	checkIfRolesAreMapped	<p>This method will call accounting service to check that for all accounting templates defined for the input product code, if the debit or credit role is derived and the role is mappable, then mapping to some GL must have been done. If all roles are mapped this service returns true, return false. </p> <p>@param productCode @param productClass @return</p>	<p>This method performs the following validations:</p> <ul style="list-style-type: none"> <li>Whether an Interest Rate Definition (IRD) is defined for the input product.</li> <li>If fail, add exception - Interest Rate Definition (IRD) not defined for the selected Product.</li> <li>Whether an Accounting Template is defined for the input product.</li> <li>If fail, add exception - Accounting Template not defined for the selected Product.</li> <li>If templates exist, system will check if there exists a corresponding GL mapping for each of the Account Roles linked in the maintenance for the input product code.</li> <li>If fail, add exception - Account Role Mapping not defined for the selected Product.</li> </ul> <p>@param product</p>
IPartyTaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyTaxAdapter	partIdMaintenance		Maintain the tax details of the primary account holder which was secondary holder of account earlier. @param accountId, oldpartyId , newpartyId
IPartyTaxAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyTaxAdapter	partIdMaintenance		If there is a change in primary ownership of account then it will update the Taxation module for the change in ownership. @param accountId @param newPrimaryPartyId
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	writeOffArrearPartially	<p>The method will be called to write off arrear partially for account.</p> <p>@param applicationContext @param accountType @param eventId @param accountId @param arrearsByTypeDTOs @param reason1 @param reason2 @param provisionDrawdownAmount @return totalWriteoffAmount</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	writeOffArrearPartially	The method will be called to write off arrear partially for account. @param applicationContext @param accountType @param eventId @param accountId @param arrearsByTypeDTOs @param reason1 @param reason2 @param provisionDrawdownAmount @return totalWriteoffAmount	This method is called to perform the partial writeoff on the account. It first calls inquireAccountWriteOff() to fetch the arrear amounts due for the account. It then performs validation that the amount of arrears balances being written off are <= the due amounts. If this validation is successful it calls writeOffArrearPartially() of CasaArrearManagementService or LoanArrearManagementService via adapters for further processing, based on the account type. For Loan account it will check that if the capital writeoff amount is greater than the principal arrear raised and due and if process date is < maturity date of the account then it will reschedule the account based on default reschedule options. @param sessionContext @param accountWriteOffDTO @return transactionStatus
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	writeOffArrearFully	The method will be called to write off arrear partially for account. @param applicationContext @param accountType @param eventId @param accountId @param reason1 @param reason2 @param provisionDrawdownAmount @return totalWriteoffAmount	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	writeOffArrearFully	The method will be called to write off arrear partially for account. @param applicationContext @param accountType @param eventId @param accountId @param reason1 @param reason2 @param provisionDrawdownAmount @return totalWriteoffAmount	This method is called to perform full writeoff on the account. @param sessionContext @param accountId @param accountType @param reason1 @param reason2 @return transactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchActiveArrearSummaryByType	This method fetches the list of all active arrears which is not paid or partially paid for the account id from running arrears table. Depending on the accountType, calls are made to respective modules to fetch arrears. @param applicationContext @param accountId @return ClassificationArrears	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchActiveArrearSummaryByType	This method fetches the list of all active arrears which is not paid or partially paid for the account id from running arrears table. Depending on the accountType, calls are made to respective modules to fetch arrears. @param applicationContext @param accountId @return ClassificationArrears	@see com.ofss.fc.app.adapter.ac.IAssetClassificationAccountAdapter#getPrincipalArrearRaisedAndDue()
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchActiveArrearSummaryByType	This method fetches the list of all active arrears which is not paid or partially paid for the account id from running arrears table. Depending on the accountType, calls are made to respective modules to fetch arrears. @param applicationContext @param accountId @return ClassificationArrears	Fetches the ClassificationArrears from Arrear Adapter @param accountId @return AccountArrearResponse @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchActiveArrearSummaryByType	This method fetches the list of all active arrears which is not paid or partially paid for the account id from running arrears table. Depending on the accountType, calls are made to respective modules to fetch arrears. @param applicationContext @param accountId @return ClassificationArrears	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchActiveArrearSummaryByType	This method fetches the list of all active arrears which is not paid or partially paid for the account id from running arrears table. Depending on the accountType, calls are made to respective modules to fetch arrears. @param applicationContext @param accountId @return ClassificationArrears	This method is called to inquire the Account Writeoff History, current arrear balances of the account and the provision amounts for the account. It internally calls fetchActiveArrearSummaryByType() method of either CasaArrearManagementService or LoanArrearManagementService based on account type to fetch the active arrear summarized by arrear type. @param sessionContext @param accountId @return accountWriteOffInquiryResponse
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	classifyAccountAsSuspended	If account classification code changes from Normal to Suspended during classification batch or manual classification then Asset Classification module will call the account module exposed methods. The arrears clawed back will also be sent to the module @param accountId @param arrear @param crrCode	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	classifyAccountAsSuspended	If account classification code changes from Normal to Suspended during classification batch or manual classification then Asset Classification module will call the account module exposed methods. The arrears clawed back will also be sent to the module @param accountId @param arrear @param crrCode	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	classifyAccountAsSuspended	If account classification code changes from Normal to Suspended during classification batch or manual classification then Asset Classification module will call the account module exposed methods. The arrears clawed back will also be sent to the module @param accountId @param arrear @param crrCode	This method updates the Account classification details to the respective account modules. @param accountAssetClassification @param actualCodeStatus
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	classifyAccountAsNormal	If account classification code changes from Suspended to Normal during classification batch or manual classification then Asset Classification module will call the account module exposed methods. @param accountId @param crrCode	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	classifyAccountAsNormal	If account classification code changes from Suspended to Normal during classification batch or manual classification then Asset Classification module will call the account module exposed methods. @param accountId @param crrCode	This method updates the Account classification details to the respective account modules. @param accountAssetClassification @param actualCodeStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	updateAccountCRRCode	This method updates the Asset Classification code or the CRR Code of the Account in the account module during batch or manual CRR change. @return void	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	updateAccountCRRCode	This method updates the Asset Classification code or the CRR Code of the Account in the account module during batch or manual CRR change. @return void	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	updateAccountCRRCode	This method updates the Asset Classification code or the CRR Code of the Account in the account module during batch or manual CRR change. @return void	This method updates the Account classification details to the respective account modules. @param accountAssetClassification @param actualCodeStatus
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	getPrincipalArrearRaisedAndDue	This method returns the amount of principal arrear raised and due for an account. @param applicationContext @param accountId @param accountCurrency @return	This method is called to perform the partial writeoff on the account. It first calls inquireAccountWriteOff() to fetch the arrear amounts due for the account. It then performs validation that the amount of arrears balances being written off are <= the due amounts. If this validation is successful it calls writeOffArrearPartially() of CasaArrearManagementService or LoanArrearManagementService via adapters for further processing, based on the account type. For Loan account it will check that if the capital writeoff amount is greater than the principal arrear raised and due and if process date is < maturity date of the account then it will reschedule the account based on default reschedule options. @param sessionContext @param accountWriteOffDTO @return transactionStatus

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchArrearWriteOffHistory	This method will fetch the account writeoff history from ArrearService. @param accountId @return	@see com.ofss.fc.app.adapter.ac.IAssetClassificationAccountAdapter#fetchArrearWriteOffHistory(java.lang.String)
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchArrearWriteOffHistory	This method will fetch the account writeoff history from ArrearService. @param accountId @return	This method will return LoanServeInquiryResponse for loanAccount .This is required for Loan Serve Make Payment Transaction. @param sessionContext @param accountId @return LoanServeInquiryResponse @throws FatalException
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchArrearWriteOffHistory	This method will fetch the account writeoff history from ArrearService. @param accountId @return	This method is called to inquire the Account Writeoff History, current arrear balances of the account and the provision amounts for the account. It internally calls fetchActiveArrearSummaryByType() method of either CasaArrearManagementService or LoanArrearManagementService based on account type to fetch the active arrear summarized by arrear type. @param sessionContext @param accountId @return accountWriteOffInquiryResponse
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			disbursements @param accountId @return	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return	The account asset classification shell will be executed only if the classification tracking is at Account level in the asset classification preference. It will be done for all accounts for which the next classification date is less than or equal to the current processing date. Once the account is picked up for processing, the next classification date is calculated and updated. Only those accounts will be considered for asset classification for which the sum of arrears is above the materiality filter limit. For accounts that do not satisfy the materiality filter check, the default asset classification code AC000 is updated as the expected code and process exits out. As of part of account classification process the following steps will be done <ul style="list-style-type: none"> <li>The expected code will be derived by executing the rule that is linked with the product class of the account. The rules will be configured in GRE based on the arrear details (DPD days for each arrear type) and other required facts.</li> <li>For account level classification process executed for Party and Facility level tracking, the expected code is updated for the account and then process picks up next account till all accounts have been processed.</li> <li>If classification applicability level is Account, the below steps are performed</li> <li>Based on the asset classification details of the current actual code that is stamped on the account and the expected code the following will be derived - Classification Event (Forward or Reverse); Classification Movement Type(Automatic or Manual) and the Classification Status(Normal-Normal, Normal-Suspended, Suspended-Suspended for Forward</li> </ul>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Event; Normal-Normal, Suspended-Normal, Suspended-Suspended for Reverse Event)</p> <ul style="list-style-type: none"> <li>If the forward or reverse movement is Automatic then the new actual code will be stamped same as the expected code.</li> <li>In case of automatic movement the asset classification accounting will be done only if status moves from Normal to Suspended or Suspended to Normal.</li> </ul> <p><u>ClassificationEventService</u></p> <p>This service will be used for classification accounting based on the Forward or Reverse event. The debit and credit amount tags will be stored in Event entity (as Day 0), for the combination of product class, event type, arrear amount type and event code. The classification event resolution service will derive the list of amount tags for the classification event request made for the product class and event type. The amount tags are populated using the arrear details of the account as follows</p> <ul style="list-style-type: none"> <li>Only arrear types that have been set for clawback in the preferences will be selected.</li> <li>For arrear types that have been capitalized, the Receivable amount tag will not be considered</li> <li>For forward event the amount to be clawed-back will either be total overdue amount or total outstanding amount as set by loan or CASA/OD module. For reverse event the amount to be clawed-back will be based on the suspended amount.</li> </ul> <p>Using the derived amount tags and other mandatory accounting attributes, the classification accounting event will be raised for the accounting module to take necessary action. The accounting module internally refers to the classification accounting template will be setup in the respective CASA and Loan products.</p> <ul style="list-style-type: none"> <li>Alerts will be raised to inform the required parties.</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p> <li>The asset classification details like classification date, expected code, actual code, expected code update date, actual code update date, reason code1 and reason code2 will be updated for the account.</li> <li>The latest asset classification details for the account will be sent to Loan and CASA/OD module for further actions.</li> <li>The next classification date is calculated and updated for the account.</li> <li>If the forward or reverse movement is Manual then the process exits after updating the expected code for the account asset classification. The actual code is updated as part of the reclassification process that is explained in <b>Inquire and Modify Account Asset Classification</b></li> <li>The account level asset classification will be done as part of batch process. As part of this process if there are any accounts that have been classified with Forward or Reverse movement as Manual then the classification update for the account will be done as part of online process via the modify operation.</li> <li>It will also be possible to inquire the existing and the history details of account-level asset classification.</li> <li>The current and history clawback details for applicable arrear types will be stored for each account.</li> <li>@param appContext @param accountData the data relating to account for which AssetClassification has to be done.</li> <li>@throws Exception @throws FatalException @throws CloneNotSupportedException</li> </p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return	This method will return LoanServeInquiryResponse for loanAccount .This is required for Loan Serve Make Payment Transaction. @param sessionContext @param accountId @return LoanServeInquiryResponse @throws FatalException
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchLoanAccountDetails	This method is responsible for fetching the LoanAccount details given an accountId. Response contains the following the category of information 1)Dates : Account Opening Date , Maturity Date , First Disbursement Date , Last Disbursement Date etc 2)Processing Information : Last Accrual Date , Last Charging Date , Last Penalty Charged Date etc. 3)Common Information : Product Code , Branch Code , Party Id , Account Currency , Number of disbursements @param accountId @return	This method is called to inquire the Account Writeoff History, current arrear balances of the account and the provision amounts for the account. It internally calls fetchActiveArrearSummaryByType() method of either CasaArrearManagementService or LoanArrearManagementService based on account type to fetch the active arrear summarized by arrear type. @param sessionContext @param accountId @return accountWriteOffInquiryResponse
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchDDAAccountDetails	This method is responsible for fetching the DDA Account details. @param accountId @return	The account asset classification shell will be executed only if the classification tracking is at Account level in the asset classification preference.  It will be done for all accounts for which the next classification date is less than or equal to the current processing date.  Once the account is picked up for processing, the next classification date is calculated and updated.  Only those accounts will be considered for asset classification for which the sum of arrears is above the materiality filter limit. For accounts that do not satisfy the materiality filter check, the default asset classification code AC000 is updated as the expected code and process exits out. As of part of account classification process the following steps will be done   <ul> <li>The expected code will be derived by executing the rule that is linked with the product class of the account. The rules will be configured in GRE based on the arrear details (DPD days for each arrear type) and other required

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>facts.</p> <ul style="list-style-type: none"> <li>For account level classification process executed for Party and Facility level tracking, the expected code is updated for the account and then process picks up next account till all accounts have been processed.</li> <li>If classification applicability level is Account, the below steps are performed -</li> <li>Based on the asset classification details of the current actual code that is stamped on the account and the expected code the following will be derived - Classification Event (Forward or Reverse); Classification Movement Type (Automatic or Manual) and the Classification Status (Normal-Normal, Normal-Suspended, Suspended-Suspended for Forward Event; Normal-Normal, Suspended-Normal, Suspended-Suspended for Reverse Event)</li> <li>If the forward or reverse movement is Automatic then the new actual code will be stamped same as the expected code.</li> <li>In case of automatic movement the asset classification accounting will be done only if status moves from Normal to Suspended or Suspended to Normal.</li> </ul> <p><u>ClassificationEventService</u></p> <p>This service will be used for classification accounting based on the Forward or Reverse event. The debit and credit amount tags will be stored in Event entity (as Day 0), for the combination of product class, event type, arrear amount type and event code. The classification event resolution service will derive the list of amount tags for the classification event request made for the product class and event type. The amount tags are populated using the arrear details of the account as follows</p> <ul style="list-style-type: none"> <li>Only arrear types that have been set for clawback in the preferences will be selected.</li> <li>For arrear types that have been capitalized, the Receivable amount tag will</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>not be considered</p> <ul style="list-style-type: none"> <li>For forward event the amount to be clawed-back will either be total overdue amount or total outstanding amount as set by loan or CASA/OD module. For reverse event the amount to be clawed-back will be based on the suspended amount.</li> <li>Using the derived amount tags and other mandatory accounting attributes, the classification accounting event will be raised for the accounting module to take necessary action. The accounting module internally refers to the classification accounting template will be setup in the respective CASA and Loan products.</li> <li>Alerts will be raised to inform the required parties.</li> <li>The asset classification details like classification date, expected code, actual code, expected code update date, actual code update date, reason code1 and reason code2 will be updated for the account.</li> <li>The latest asset classification details for the account will be sent to Loan and CASA/OD module for further actions.</li> <li>The next classification date is calculated and updated for the account.</li> <li>If the forward or reverse movement is Manual then the process exits after updating the expected code for the account asset classification. The actual code is updated as part of the reclassification process that is explained in <b>Inquire and Modify Account Asset Classification</b></li> <li>The account level asset classification will be done as part of batch process. As part of this process if there are any accounts that have been classified with Forward or Reverse movement as Manual then the classification update for the account will be done as part of online process via the modify operation.</li> <li>It will also be possible to inquire the existing and the history details of account-level asset classification.</li> <li>The current and history clawback</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				details for applicable arrear types will be stored for each account.  @param appContext @param accountData the data relating to account for which AssetClassification has to be done. @throws Exception @throws FatalException @throws CloneNotSupportedException
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchDDAAccountDetails	This method is responsible for fetching the DDA Account details. @param accountId @return	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchDDAAccountDetails	This method is responsible for fetching the DDA Account details. @param accountId @return	This method is called to inquire the Account Writeoff History, current arrear balances of the account and the provision amounts for the account. It internally calls fetchActiveArrearSummaryByType() method of either CasaArrearManagementService or LoanArrearManagementService based on account type to fetch the active arrear summarized by arrear type. @param sessionContext @param accountId @return accountWriteOffInquiryResponse
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	doReschedule	This method calls LoanRepaymentScheduleApplicationService.doReschedule() LoanRepaymentScheduleApplicationService.generateRepaymentRecordsOnReschedule() LoanRepaymentScheduleApplicationService.saveScheduleOnReschedule() @param sessionContext @param accountId @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param	This method calls LoanRepaymentScheduleApplicationService.doReschedule() LoanRepaymentScheduleApplicationService.generateRepaymentRecordsOnReschedule() LoanRepaymentScheduleApplicationService.saveScheduleOnReschedule() @param sessionContext @param accountId @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param reasonCode @param reasonDesc @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			reasonCode @param reasonDesc @return	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	doReschedule	This method calls LoanRepaymentScheduleApplicationService.doReschedule() LoanRepaymentScheduleApplicationService.generateRepaymentRecordsOnReschedule() LoanRepaymentScheduleApplicationService.saveScheduleOnReschedule() @param sessionContext @param accountId @param rescheduleDate @param rescheduleOption @param capitalizeArrears @param reasonCode @param reasonDesc @return	This method is called to perform the partial writeoff on the account. It first calls inquireAccountWriteOff() to fetch the arrear amounts due for the account. It then performs validation that the amount of arrears balances being written off are <= the due amounts. If this validation is successful it calls writeOffArrearPartially() of CasaArrearManagementService or LoanArrearManagementService via adapters for further processing, based on the account type. For Loan account it will check that if the capital writeoff amount is greater than the principal arrear raised and due and if process date is < maturity date of the account then it will reschedule the account based on default reschedule options. @param sessionContext @param accountWriteOffDTO @return transactionStatus
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchDDAccountPreferences	@param sessionContext @param accountId @return @throws FatalException	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchDDAccountPreferences	@param sessionContext @param accountId @return @throws FatalException	This method is called to inquire the Account Writeoff History, current arrear balances of the account and the provision amounts for the account. It internally calls fetchActiveArrearSummaryByType() method of either CasaArrearManagementService or LoanArrearManagementService based on account type to fetch the active arrear summarized by arrear type. @param sessionContext @param accountId @return accountWriteOffInquiryResponse



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	getTransactionAmountSDTO	This method returns TransactionAmountsAdapterDTO which holds amountInTransactionCurrency, amountInAccountCurrency and transactionToAccountRate @param transactionAmount @param accountCurrency @return	This method returns TransactionAmountsAdapterDTO which holds amountInTransactionCurrency, amountInAccountCurrency and transactionToAccountRate @param transactionAmount @param accountCurrency @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	fetchPaidArrearsTillDate	This service fetches all the arrears paid till date. A @param accountId @return	This method fetched various account balances like Outstanding balance, Approved limit, Interest accrued, Interest and fee recovered. @param sessionContext @param accountId @param moduleType @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	isAccountNominatedForBundleFee	@param accountId @param accountType @return	
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	isAccountInCollection	The method is responsible to inquire from account module if an account is present in collection. @param accountId @param moduleType @return	The method is responsible to inquire from account module if an account is present in collection. @param accountId @param moduleType @return
IAssetClassificationAccountAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ac\IAssetClassificationAccountAdapter	isAccountInCollection	The method is responsible to inquire from account module if an account is present in collection. @param accountId @param moduleType @return	This method triggers accounting event, updates Written Off recovered amount and inserts Written Off recovery
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchPartyBasicDetails		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchPartyBasicDetails		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchPartyBasicDetails		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllAccountsForSchedule		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllAccountsForSchedule		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllAccountsForSchedule		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllPartiesOfAccount		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllPartiesOfAccount		This method is used to call the service PartyAccountRelationshipApplicationService.fetchAllPartiesOfAccount(SessionContext, AccountId) which is used to return the details details of all the parties for a given account along with the relationship that exists @param accountId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchAllPartiesOfAccount		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	validateNickName		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	validateNickName		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	validateNickName		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	validateNickName		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchNameAndRelationShipForParty		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchNameAndRelationShipForParty		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchNameAndRelationShipForParty		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	fetchNameAndRelationShipForParty		
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	verifyDirectEntryCode	This method will fetch the Direct Entry Identification Number for the provider partyId.  @param partyId @param direction @param inputDirectEntryCode @return	This method will validate the Direct Entry Identification Number for the provider account.  @param accountCode @param direction @param inputDirectEntryCode
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	verifyDirectEntryCode	This method will fetch the Direct Entry Identification Number for the provider partyId.  @param partyId @param direction @param inputDirectEntryCode @return	
IPCPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\pc\IPCPartyAdapter	verifyDirectEntryCode	This method will fetch the Direct Entry Identification Number for the provider partyId.  @param partyId @param direction @param inputDirectEntryCode @return	
IDDAFTPAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAFTPAdapter	calculateQuoteExpiryDate	Method will be used to calculate quote expiry date. @param quoteGenerationDate @param accountId @return quoteExpiryDate	Method will be used to calculate quote expiry date. @param quoteGenerationDate @param accountId @return quoteExpiryDate
IDDAFTPAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAFTPAdapter	calculateQuoteExpiryDate	Method will be used to calculate quote expiry date. @param quoteGenerationDate @param accountId @return quoteExpiryDate	Method will return quote expiry date for fixed period OD quote. @param accountId @param quotePeriod @param quoteGenerationDate @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireProduct	This method used to fetch product details . @param productId	This method is used for validating account type whether it is of type Asset.If above condition is not satisfied,validation error is thrown. @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireProduct	This method used to fetch product details . @param productId	Method will populate details of BasicInquiryDetailsDTO using entity {@link IDemandDepositAccount}; @param demandDepositAccount @return BasicInquiryDetailsDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireProduct	This method used to fetch product details . @param productId	This method is used to inquire the savingspreferencesApplicationService :: fetchSavingspreferences so as to populate all the details required to set different preferences at the time of account opening @param productId @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductPreferencesForHIOSA	This method fetches the product preferences which indicates whether funds can be transfered to which kind of account. @param productId @return	This method check whether transfer between these accounts is allowed or not. If toAccount is null then it is treated as external account. @param fromAccount @param fromAccountType @param toAccount @param toAccountType @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductPreferencesForHIOSA	This method fetches the product preferences which indicates whether funds can be transfered to which kind of account. @param productId @return	This method check whether transfer between these accounts is allowed or not. If toAccount is null then it is treated as external account. @param fromAccount @param fromAccountType @param toAccount @param toAccountType @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	This method validates the personalization option. Personalization option can be selected if the personalization option is set at the product level. The option of selecting the nick name is valid if the personalization option is valid. @param request @param preferencesDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	This method fetches the cheque preferences set at the product level. @param account @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	This method fetches the number of cheques requested in the input period for the accountId and the cheque book type. This method calls <code>ChequeManagementService#getCountOfChequesRequestedInAPeriodByType</code> @param accountId @param startDate @param endDate @param chequeBookType @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	@see <code>com.ofss.fc.app.dda.service.core.IDemandDepositInstructionService#requestNewChequeBook(com.ofss.fc.app.context.SessionContext, java.lang.String)</code>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreference	This method fetches the product cheque book preferences. @param productId @return	This method will override chequebook preferences for new offer. @param oldOfferId @param newOfferId @param demandDepositAccount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductOfferAndPreference	This method used to fetch casa and offer product preferences @param productId	This method is used to obtained the product adapter @param demandDepositAccount @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	This method will populate basic facts for casa account. @param bankingFactContextDTO @param accountDTO @return bankingFactContextDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	Maximum number of accounts that a party can have under the product as the primary (first) holder will be defined at the Product level. System should not permit CASA account opening if this account is breaching the set limit. System should permit CASA account opening only if the product code is active and not expired. It works as follows: <li>Get the count of account for which the party is primary account holder by calling IDDAPartyAdapter#fetchCountPrimaryAccountHolder(partyId : String)</li> <li>Compare the count with the allowable count at the product level.</li> <li>Reject account opening if the product level count is less than or equal to the count for which party is primary account holder.</li> @param product
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	In Manage facility, Using existing facility , New OD account is created.This method is used to create new recommendation while opening account.Following are steps involved. <ol> <li>Create new Price ticket. <li>Create facts based on new attributes. <li>Create new recommendation. </ol> @param sessionContext @param pricingRequestDTO @param factMap @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	Method will be used to validate all the facilities {@link BankingFacilities} and {@link AccountFacilities} provided to account with Product defined under.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	This method is used to fetch the details from product module based on product Id Thedata fetched from the product module will be used for cloning the account @param productCode @return DDAProductPreferencesDTO @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferAndProductPreference	This method used to fetch casa and offer product preferences @param productId	This method is used to obtained the product adapter @param demandDepositAccount @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchFunds Acquisition Details	Every product has its own funds utilization sequence defined. For funds utilization this sequence need to be followed. This method provides the funds utilization or funds acquisition sequence defined for the product. @param productId @return	This method derives the product details which are required in deriving the funds acquisition sequence and funds apportionment sequence. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchFunds Acquisition Details	Every product has its own funds utilization sequence defined. For funds utilization this sequence need to be followed. This method provides the funds utilization or funds acquisition sequence defined for the product. @param productId @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<pre> This method will validate the product code passed by the Origination module during account opening. The following is the algorithm for the method: &lt;li&gt;Validate if the product code exists or not. &lt;/li&gt; &lt;li&gt;Call {@link IDDAProductAdapter#inquireProduct}&lt;/li&gt; &lt;li&gt;If product exist then &lt;li&gt;call {@link this.validateProductExpiry(product: Product)} to check if product is not expired&lt;/li&gt; &lt;li&gt;call {@link this.validatePermissibleNoOfAccounts(product: Product)} to check if this party has not breached the max no of account limit&lt;/li&gt; &lt;li&gt;call {@link this.validateRestrictions(product.getKey().keyAsString())}&lt;/li&gt; &lt;/li&gt; &lt;/pre&gt; </pre>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<pre> This method is exposed for performing various business policy validations. This will internally call all the business policy validation methods and return an array of {@link ValidationError} in case of errors. &lt;/pre&gt; @throws FatalException </pre>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<p>&lt;ol&gt;The method validatePolicy() is by default called by the validate() method of the AbstractBusinessPolicy class. &lt;li&gt;This method first call the validate policy method of the parent class to call the common validations&lt;/li&gt; &lt;li&gt;Then give call to other methods in the business policy&lt;/li&gt; &lt;/ol&gt;</p>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<p>This method will validate the Product combination for Fund Transfer. Load the Account and get the productId. Call {@link com.ofss.fc.app.adapter.dda.IDDAProductAdapter.isValidProductCombinationforEvent(String, String, String)}. If returns true, product combination is valid else product combination is not valid</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	The method validate if the product expiry date is not less than the current posting date.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<p>This method will be called from service to validate the business policy for offer swap event. </p> @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	<p>This method checks if the old offer and the new offer belong to the same product. If they do not, then it throws a business exception &#x2013;  </p> <ol> <li>FC_DD_OS_012, Offer Swap cannot be performed. Invalid UDM option. <li>FC_DD_OS_013, Offer Swap cannot be performed. Old offer {0} on the account and new offer {0} belong to different products. </ol> @param DDAOfferDTOMap
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	This method is used to check whether the product is valid or not by inquiring the product module It checks whether the given product is in active state at the product module side or not, In case if it active it returns true else returns false @param sessionContext @param productCode @return boolean
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProduct	Check for valid Product	The method validate if the product expiry date is not less than the current posting date. Get the postingDate from the session context. product object is in input. If (product.DateRange.endDate != null && product.DateRange.endDate < postingDate ) then add this as a ValidationError as product is expired. FALSE value indicate that the product is valid product and not expired. TRUE value indicate that the product cannot be used and is invalid. @param product

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProductExpiry	Check for product expiry	<pre> This method will validate the product code passed by the Origination module during account opening. The following is the algorithm for the method: &lt;li&gt;Validate if the product code exists or not. &lt;/li&gt; &lt;li&gt;Call {@link IDDAProductAdapter#inquireProduct}&lt;/li&gt; &lt;li&gt;If product exist then &lt;li&gt;call {@link this.validateProductExpiry(product: Product)} to check if product is not expired&lt;/li&gt; &lt;li&gt;call {@link this.validatePermissibleNoOfAccounts(product: Product)} to check if this party has not breached the max no of account limit&lt;/li&gt; &lt;li&gt;call {@link this.validateRestrictions(product.getKey().keyAsString())}&lt;/li&gt; &lt;/li&gt; </pre>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	validateProductExpiry	Check for product expiry	<p>The method validate if the product expiry date is not less than the current posting date. Get the postingDate from the session context. product object is in input. If (product.DateRange.endDate != null &amp;&amp; product.DateRange.endDate &lt; postingDate ) then add this as a ValidationError as product is expired. FALSE value indicate that the product is valid product and not expired. TRUE value indicate that the product cannot be used and is invalid. @param product</p>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductIRDMapping	This method is used to fetch the Product related IRD's details(Interest rule definition)	<p>This method is used to populate the values for InterestRegistrationRequestDTO @param dDAInterestRegistrationDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductIRDMapping	This method is used to fetch the Product related IRD's details(Interest rule definition)	This method performs the Interest Settlement according to the Interest Class defined at the Product Manufacturing Unit for the respective product attached to the Account being closed. It interacts with Product Manufacturing Unit and fetches all Product-Interest Rule mapping details defined for the particular Product. - If Interest Class defined for the Product is CREDIT, then Credit Interest Settlement is performed. - If Interest Class defined for the Product is DEBIT, then Credit Interest Settlement is performed. - If Interest Class defined for the Product is CREDIT and DEBIT, then both Credit & Debit Interest Settlement is performed. @param applicationContext @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductIRDMapping	This method is used to fetch the Product related IRD's details(Interest rule definition)	When account status changes from <b> Dormant </b> to <b> Unclaimed </b>, credit and debit Settlement action takes place.   Following are the key points. <ol> <li>This is adhoc settlement similar to account closure. <li>Credit as well as Debit settlement(if applicable) will be performed. <li>Interest will be computed till process date. <li>If account is in debit balance , account will not be marked as unclaimed. Idea is to recover all the funds before marking status as unclaimed. <li>Once account is unclaimed, we can not perform any transaction. </ol> @param applicationContext @param demandDepositAccount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method prepares {@link DocumentDTO} which is required for report operation.This contains the documentTypeId used from {@link DemandDepositConfiguration} file. It also contains the productGroup. @param demandDepositAccount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method fetches the product group nature of a product code. @param productCode @param fieldName @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method uses IDDAProductAdapter and fetches productGroupId for the product. @param productId @return productGroupId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method prepares {@link DocumentDTO} which is required for report operation. This contains the documentTypeId used from {@link DemandDepositConfiguration} file. It also contains the productGroup. @param demandDepositAccount @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method prepares {@link DocumentDTO} which is required for report operation. This contains the documentTypeId used from {@link DemandDepositConfiguration} file. It also contains the productGroup. @param demandDepositAccount @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method prepares {@link DocumentDTO} which is required for report operation. This contains the documentTypeId used from {@link DemandDepositConfiguration} file. It also contains the productGroup. @param demandDepositAccount @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method fetches the product group of the input product code. @param productCode @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroup	This method fetches the product group of the input product code. @param productCode @return @throws FatalException	This method fetches the product group nature of the input product group. @param productGroup @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductGroupNature	This method fetches the product group nature of the input product group. @param productGroup @return @throws FatalException	This method fetches the product group nature of a product code. @param productCode @param fieldName @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method validates the alternate credit account. <ol> If account type is DDA <ol> Check if required data is present in the map .If not fetch the details and validate </ol> <li>Existence of account</li> <li>Product Group Nature (Should be asset or liability)</li> <li>System defined status of account (Only regular and inactive are valid status)</li> <li>User defined status of account (Should not have any user defined status)</li> </ol> </ol> <ol> If account type is Loan, validate the following <li>Existence of account</li> <li>Status of the account.( Should be regular)</li> </ol> In case of any validation error, appropriate validation error is added to the list of validation errors. The details of the account are added to the map and the map is returned. @param parentId @param accountIdForCreditPosting @param accountType @param map @return HashMap

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method validates the alternate debit account. <ol> Check if required data is present in the map .If not fetch the details and validate </ol> <li>Existence of account</li> <li>Product Group Nature (Should be asset)</li> <li>System defined status of account (Only regular and inactive are valid status)</li> <li>User defined status of account (Should not have any user defined status)</li> <li>Currency code is same as that of parent account</li> </ol> In case of anyValidation error, appropriate validation error is added to the list of validation errors. The details of the account are added to the map and the map is returned. @param parentAccountId @param accountIdForDebitPosting @param accountType @param map @return HashMap
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method validates the alternate charges account. <ol> Check if required data is present in the map .If not fetch the details and validate </ol> <li>Existence of account</li> <li>Product Group Nature (Should be asset)</li> <li>System defined status of account (Only regular and inactive are valid status)</li> <li>User defined status of account (Should not have any user defined status)</li> <li>Currency code is same as that of parent account</li> </ol> In case of anyValidation error, appropriate validation error is added to the list of validation errors. The details of the account are added to the map and the map is returned. @param parentAccountId @param accountIdForChargesPosting @param accountType @param map @return HashMap
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method fetches the product group nature of a product code. @param productCode @paramfieldName @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method prepares the response using the Account Balances available for the provided Account. @param transactionDates @param demandDepositAccount @param accountBalancesMap @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method converts prepare AccountFeederWrapperDTO with mandatory values. @param accountId @return AccountFeederWrapperDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchProductClass	This method fetch the Product class of the product which indicates whether the product is of type Savings or Checkings. @param productId @return	This method will fetch the product class of the respective product code i.e. whether it is current or savings and set it in casa internal product class of ProductClassInfoDTO ,it will also set the product code as CASA as returned from PMU depending on product code. Hence if you require the casa internal product code such as CH or CS you need to fetch it from the productClassInfoDTO.getCasaInternalProductClass if you require the product class as defined in PMU then call productClassInfoDTO.getProductClass @param productId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreferences		
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreferences		This method fetches the cheque preferences set at the product level. @param account @return



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreferences		This method fetches the number of cheques requested in the input period for the accountId and the cheque book type. This method calls <code>ChequeManagementService#getCountOfChequesRequestedInAPeriodByType</code> @param accountId @param startDate @param endDate @param chequeBookType @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreferences		@see <code>com.ofss.fc.app.dda.service.core.IDemandDepositInstructionService#requestNewChequeBook(com.ofss.fc.app.context.SessionContext, java.lang.String)</code>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAProductChequePreferences		This method will override chequebook preferences for new offer. @param oldOfferId @param newOfferId @param demandDepositAccount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isValidProductCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @param eventId @return	This method will validate the Product combination for Fund Transfer. Load the Account and get the productId. Call {@link <code>com.ofss.fc.app.adapter.dda.IDDAProductAdapter.isValidProductCombinationforEvent(String, String, String)</code> }. If returns true, product combination is valid else product combination is not valid

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isValidProductCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @param eventId @return	This method fetches the product combination validity. @param map @param parentAccountId @param alternateAccountId @param fieldName
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isValidProductCombinationforEvent	This method validates with the HISO restriction maintained at product level, whether the funds can be transfer from from account to to account. If the restriction is not maintained between these accounts then, the transfer is allowed, else this transfer is not allowed. @param fromProductId @param toProductId @param eventId @param eventId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isDormancyRuleUsed	This method validates the dormancy rule maintained at product level. @param dormancyRuleId @return	This method is exposed to perform delete operation for AgingRuleConfiguration @param agingRuleConfiguration
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isDormancyRuleUsed	This method validates the dormancy rule maintained at product level. @param dormancyRuleId @return	This method validates the dormancy rule maintained at product level. @param dormancyRuleId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isAufFacilityAvailableOnProduct	This method checks whether AUF facility is available on product or not. @param productId @return	This method checks all the validations the business rules that applies on the auf limit setup. For Example : <ul> <li>AUF Limit can only be altered if the account status allows. <li>AUF Limit can only be modifies to an extent of utilization, it cannot be decreased that auf limit utilization.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	deriveInterestCalculationNature	The method find out if the product belonging to the account is a collected balance type or a book balance type. <ol> <li>Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).</li> <li>Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.</li> </ol> There is a processing dependency on dis-honour of an outward clearing. <ol> <li>If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.</li> <li>if the product is of the collected balance then dishonour outward record will be updated as of current value date.</li> </ol> The alogirthm for deciding the interest calculation nature should be as follows: <ul> <li>Get the Balance	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>base type and expression id by calling a service in the interest module passing the credit IRD as input.</p> <ul style="list-style-type: none"> <li>&lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collected Balance.</li> <li>&lt;li&gt;If balance base is Fact based, then nature = InterestCalculationNature.Collected Balance.</li> <li>&lt;li&gt;If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression</li> <li>&lt;li&gt;If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;</li> <li>&lt;li&gt;If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collected Balance;</li> <li>&lt;li&gt; If balance base type is FACT or Rule then return false.</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveInterestCalculationNature	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;Get the Balance base type and expression id by calling a service in the interest module passing the credit IRD as input.&lt;/li&gt; &lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collecte</p>	<p>This method will prepare the accounting entries for Recurring instruction execution. @param sourcingResponse @param executionResponse @param amountTag @return</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Balance.</p> <ul style="list-style-type: none"> <li>If balance base is Fact based, then nature = InterestCalculationNature.Collecte</li> </ul> <p>Balance.</p> <ul style="list-style-type: none"> <li>If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression</li> <li>If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;</li> <li>If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collecte</li> </ul> <p>Balance;</p> <p>If balance base type is FACT or Rule then return false.</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveInterestCalculationNature	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;Get the Balance base type and expression id by calling a service in the interest module passing the credit IRD as input.&lt;/li&gt; &lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collecte</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Balance.</p> <ul style="list-style-type: none"> <li>If balance base is Fact based, then nature = InterestCalculationNature.Collecte</li> <li>Balance. If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression</li> <li>If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;</li> <li>If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collecte</li> <li>Balance;</li> <li>If balance base type is FACT or Rule then return false.</li> </ul>	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveInterestCalculationNature	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;Get the Balance base type and expression id by calling a service in the interest module passing the credit IRD as input.&lt;/li&gt; &lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collecte</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Balance.</p> <ul style="list-style-type: none"> <li>If balance base is Fact based, then nature = InterestCalculationNature.Collecte</li> <li>Balance. If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression</li> <li>If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;</li> <li>If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collecte</li> <li>Balance;</li> <li>If balance base type is FACT or Rule then return false.</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveInterestCalculationNature	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;Get the Balance base type and expression id by calling a service in the interest module passing the credit IRD as input.&lt;/li&gt; &lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collecte</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Balance.</p> <ul style="list-style-type: none"> <li>If balance base is Fact based, then nature = InterestCalculationNature.Collecte</li> <li>Balance. If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression</li> <li>If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;</li> <li>If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collecte</li> <li>Balance;</li> <li>If balance base type is FACT or Rule then return false.</li> </ul>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveInterestCalculationNature	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithhm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;Get the Balance base type and expression id by calling a service in the interest module passing the credit IRD as input.&lt;/li&gt; &lt;li&gt;If balance base is Rule based, then nature = InterestCalculationNature.Collecte</p>	<p>The method find out if the product belonging to the account is a collected balance type or a book balance type. &lt;ol&gt; &lt;li&gt;Book Balance type : If uncleared funds IS used in the balance base expression, then it is called book balance (because we are adding uncleared funds to the clear funds and calculating interest on it).&lt;/li&gt; &lt;li&gt;Collected Balance type: If uncleared funds is NOT used in the balance base expression, then it is called collected balance.&lt;/li&gt; &lt;/ol&gt; There is a processing dependency on dis-honour of an outward clearing. &lt;ol&gt; &lt;li&gt;If the product is of type book balance then dishonour outward record will update the casa balance as of the value date on which the uncleared funds were increased.&lt;/li&gt; &lt;li&gt;if the product is of the collected balance then dishonour outward record will be updated as of current value date.&lt;/li&gt; &lt;/ol&gt; The algorithhm for deciding the interest calculation nature should be as follows: &lt;ul&gt; &lt;li&gt;call the ProductManufacturingApplicationService to get casaPreferences details. &lt;/li&gt; &lt;li&gt;If casaPreferences.sCollectedBalanceEnabled is true, then nature = InterestCalculationNature.Collecte &lt;li&gt;else, nature = InterestCalculationNature.BookBalance. &lt;/li&gt;&lt;/ul&gt;</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<p>Balance.&lt;/li&gt; &lt;li&gt;If balance base is Fact based, then nature = InterestCalculationNature.Collected Balance. &lt;li&gt;If balance base type is expression then call com.ofss.fc.domain.rule.service.expression.ExpressionService.getFactsUsedInExpression(String) to get all the facts used in the expression &lt;li&gt;If any of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.BookBalance;&lt;/li&gt; &lt;li&gt;If non of the fact matches with unclear funds then iInterest calculation nature of the product is InterestCalculationNature.Collected Balance;&lt;/li&gt; If balance base type is FACT or Rule then return false.</p>	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isProductClassCurrent	<p>Check the product class from pmu module @param productId @return</p>	<p>This method will fetch the product class of the respective product code i.e. whether it is current or savings and set it in casa internal product class of ProductClassInfoDTO ,it will also set the product code as CASA as returned from PMU depending on product code. Hence if you require the casa internal product code such as CH or CS you need to fetch it from the productClassInfoDTO.getCasaInternalProductClass if you require the product class as defined in PMU then call productClassInfoDTO.getProductClass @param productId</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isProductClassCurrent	Check the product class from pmu module @param productId @return	This method returns true or false depending on the product class from product manufacturing account is current or savings @param productCode @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchUDMDefinition	This method will be used to inquire UDM definition maintained at product level.  If UDM is not maintained in PMU then it will simply return null. @param productId @return	This method will be used to execute bonus interest processing.  Sequence for execution-  <ol><li>Inquire product id by loading DDAccount.<li>check for UDM maintenance at product level if present then <li>Get all the facts from consulting to evaluate the rule. <li>Validate rule for bonus interest eligibility and if account is eligible for bonus interest then <li>log the non financial statement item with description. <li>check if bonus interest is applicable for current cap cycle then call Interest module to get the start date and end date of current cycle and Call pricing to create back dated UDM Definition at account level, provide start date and end date also.  and instead of logging backdated action process backdated right away. <li>And if bonus interest is applicable for next cap cycle then create future dated UDM with start and end date and log the rate change action with due date as start date. </ol> @param applicationContext Application Context which will be set in batch process. @param actionKey @param actionRequest every subclass need to cast this to appropriate subclass of {@link ActionKey} @param dueDate due date for the action
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAOfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	The following method is used to fetch account related information using the account type.The following method is used to avoid interaction begin and close related calls. @param accountId @param accountType @param response @return @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	@param offerId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	The Minimum Balance Amount should get Restored after completion of EOD of the Valid Till date and the valid till date also should get restored as it was before modification. Reset of amount will be done as per the currency attributes user product preference. @param demandDepositAccount TODO @return the minimumBalanceRequired
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	This method checks whether AUF facility is available on product or not.  @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	This method fetches the product cheque book preferences. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA OfferInquiryResponse	This method is used to fetch the offer details. @param offerId @return	This method fetches the product preferences which indicates whether funds can be transferred to which kind of account. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getOfferName	This method is used to fetch the Offer name with respect to offerId. @param offerId @return	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getOfferName	This method is used to fetch the Offer name with respect to offerId. @param offerId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getOfferName	This method is used to fetch the Offer name with respect to offerId. @param offerId @return	Method will populate details of BasicInquiryDetailsDTO using entity {@link IDemandDepositAccount}; @param demandDepositAccount @return BasicInquiryDetailsDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getOfferName	This method is used to fetch the Offer name with respect to offerId. @param offerId @return	This method will fetch the offer name for a given offer id by calling the OfferApplicationService @param offerId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getOfferName	This method is used to fetch the Offer name with respect to offerId. @param offerId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method verify whether offer is expired or not.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method verify whether offer start date is greater than backdated account opening date. <b>Since:</b>R2.2  
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method verify whether offer is expired or not.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	<p>This method will be called from service to validate the business policy for offer swap event. </p> @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	<p>This method makes a call to the dda product adapter and fetches all details for old and new offer and then put them into map. </p> @param newOfferId @param oldOfferId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	<p>This method will update the account facilities to those present in the new offer. The facilities updated here are: <ul> <li>Sweep In</li> <li>Sweep Out</li> <li>Standing Instruction</li> </ul> </p> @param newOfferId @param demandDepositAccount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method will fetch PricingTicketNumber and generate recommendation. @param offerSwapDetails @return String @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method will override chequebook preferences for new offer. @param oldOfferId @param newOfferId @param demandDepositAccount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method is used to fetch the casa product details from the CASAProductPreferenceApplicationService. @ param offerId @ param currencyCode
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method would return the offer Inquiry response

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method is used to populate the values for DDAProductValidationDTO @param productDetailsDTO @param savingspreferencesDTOResponse
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param currencyCode @param offerId @return reqMinimumAmt
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method returns whether personalised flag is checked for the offer id passed as parameter. @param offerId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method derives the singleStatementEntry flag for given account id.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchOfferDetails	This method will fetch the offer detail for an offer id @param offerId @return	This method will whether fee for MFD transaction is defined.Returns true if fee to be charged is at MFD level.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireCASAEconomicCostDetails	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO	Method will be used to inquire Economics Cost details maintained at product level. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireCASAEconomicCostDetails	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO	Method will be used to inquire Economics Cost details maintained at product level. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireCASAEconomicCostDetails	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO	Method will be used to inquire Economics Cost details maintained at product level. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireCASAEconomicCostDetails	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO	Method will be used to inquire Economics Cost details maintained at product level. @param productId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	inquireCASAEconomicCostDetails	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO	Method will be used to inquire Economic cost details. @param productId @return costDetailsDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAPreferencesInquiryResponse	@param productId @return	This method is used to validate the simulation date

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	On account opening, method id called to mark the fees as deferred or capitalised. Following steps are performed. <ol> <li>Get the outlay funds which are stored against recommendation pricing ticket.Refer DDAOpeningApplicationService.inquireServicingFees ForRecommendation. <li>Mark fees as deferred or capitalised in pricing module. <li>Store outlay funds against account number. </ol> @param pricingTicketNumber @param accountId @param productId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	On Amend OD workflow, After inquiring fees we mark the fees as cap or deferred and update account id for fee ticket number. @param odAmendConfigurationDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	Fetch CASA product preferences. @param productCode @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	This method works as: <ol> <li>If principal offset is allowed and selected as variable rate period then begin date will be end date and its next date. <li>If principal offset is not allowed and selected as variable rate period then begin date will be current process date. <li>If principal offset is allowed and selected as fixed rate period then begin date will be current process date. <li>If principal offset is allowed and selected as fixed rate period then end date will be current date. <li>If allowed rate type is other than variable or fixed then method returns null. </ol> @param demandDepositAccount @param beneficiaryAccount @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	Credit payments are allowed on asset type of accounts although account is closed or written off. Method returns true when <ol> <li>Account is closed or written off. <li>Account is asset. <li>Product level flag for Accept payment for closed /written off account is true. <li>Service.isAllowedOnWrittenOffAccounts is true. @param accountId @param eventId @param serviceId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	This method will validate the existing DemandDepositAccount Appropriation Sequence with the new Appropriation Sequence,if the existing Appropriation Sequence is not equal to updated Appropriation Sequence then add to existing validation error. @param controlParametersDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	Method prepares and stores servicing fee response. @param accountServicingFeeRequestDTO @param priceFeaturesDTOs @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASA Preferences InquiryResponse	@param productId @return	Method prepares and stores servicing fee response. @param accountServicingFeeRequestDTO @param priceFeaturesDTOs @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	This method is responsible for charging MinimumAverageBalance fee if average balance is not maintained in the account. Assumptions: <ul><li>BalanceChangeAction will execute before MinimumAverageBalance action</li> <li>Fee collection for account closure will be handled by account closure service</li> </ul>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	Account preferences can be changed after account opening using account preferences screen. This method used to log action after checking new minimum balance preferences. @param domainObject

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	This method is responsible for computing average balance for current period. Assumptions: <ol><li>Minimum average balance at account level will be 0 if no fees need to be charged.</li> <li>Fees will be charged on previous working day if last of the cycle is holiday.</li> </ol> @param valueDate Value date will be one of the following depending on the eventtype: <ul> <li>Account opening date</li> <li>Value date on which balance has changed.</li> <li>Account closure date</li> </ul> @param averageBalanceEventType Eventtype @param closingBalance Closing balance at the time of AccountClosure @param balanceHistoryItem For backdated entry, this will be used for determining Current, OD & TOD balances on that effective date. return Represents average balance for current period or applicable fee for AccountCloseInquiry
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	This method returns Minimum balance. It works as follows: 1) Returns if MinimumBalanceType is NONE 2) Returns balance required set at account level if valueDate is within validity period 3) Returns balance required set at product level



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	This method merges existing account preferences with the preferences captured during account configuration @param newPreferences @param existingPreferences @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	getMinimumBalanceRequired	This method will be used to inquire minimum balance required for particular product. This method will only return minimum balance required nothing else and written specifically for batch call as a performance fix.  It will be called from MinimumAverageBalanceHelper.getMinimumBalanceRequired(DemandDepositAccount). @param productId @param offerId @return reqMinimumAmt	This method is used to set the minimum balance preferences which is already obtained from com.ofss.fc.domain.dda.service.account.core.DDAccountOpeningService.fetchProductDetailsForCloneAccount(String, String) Minimum balance is required for com.ofss.fc.domain.dda.entity.account.core.DemandDepositAccount.initializeAverageBalances(Date), hence setting the attribute values @param minimumBalancePreferencesAdapter @param openingDate @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCasaProductDormancyRule	This method will be used to inquire dormancy rule for CASA product. This has been added as a performance enhancement to avoid flush in batch. @param productId @return dormancyRuleId	Method will return object of AgingRuleConfiguration. @param productId @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCasaProductDormancyRule	This method will be used to inquire dormancy rule for CASA product. This has been added as a performance enhancement to avoid flush in batch. @param productId @return dormancyRuleId	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method add the validations error for td related events @param oldDDAOfferDTO @param newDDAOfferDTO @param tdFromOfferId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method would validate the new offer id should not be same as the restriction maintained for TD maturity where the account number passed is beneficiary account number in TD principal payout setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method would validate the new offer id should not be same as the restriction maintained for Loan disbursement where the account number passed is beneficiary account number in Loan disbursement setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method add the validations error for Periodic funds transfer related events @param oldDDAOfferDTO @param newDDAOfferDTO @param toOfferId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method add the validations error for Sweep Out setup related events @param oldDDAOfferDTO @param newDDAOfferDTO @param toOfferId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method would validate the new offer id should not be same as the restriction maintained for Periodic Funds Transfer where the account number passed is beneficiary account number in Periodic funds transfer setup @param oldDDAOfferDTO @param newDDAOfferDTO @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchToOfferList	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service	This method would return the list of to offer id's for given event code and from offer id from the product manufacturing service
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isTemporaryLimitNotConsiderable	This method will tell id the temporary excess limit to be considered for deriving the net withdrawble amt.	Method will populate details of BalanceInquiryDetailsDTO using entity {@link IDemandDepositAccount}; @param demandDepositAccount @return BalanceInquiryDetailsDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchAdditionalFacilities	This method will return the list of AdditionalFacilityDTO for given offerId.	This method set AdditionalFacilities for an account.It fetch AdditionalFacilities by passing offerId as param and return ddaProductPreferencesDTOs.And it convert to additional facilities.if additional facilities is null then it set to demandDepositAccount. @param demandDepositAccount @param productPreferencesDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH/AT. @param eventId @return @throws FatalException	This method will call the checkForNotNull method to check the Null inputs and then set the different account transaction attributes required for transaction. @param accountTransaction @param sourceCurrency @param accountCurrencyCode @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH/AT. @param eventId @return @throws FatalException	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH/AT. @param eventId @return @throws FatalException	This method is used call the product service to fetch domain category from the given event code. @fcb.param MI,DemandDepositCashTransactionService,@param eventId @fcb.param MI,DemandDepositCashTransactionService,@return @fcb.param MI,DemandDepositCashTransactionService,@throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH/AT. @param eventId @return @throws FatalException	This method is used call the product service to fetch domain category from the given event code. @fcb.param MI,DemandDepositCashTransactionService,@param eventId @fcb.param MI,DemandDepositCashTransactionService,@return @fcb.param MI,DemandDepositCashTransactionService,@throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDomainCategoryFromEvent	This method would call the product service to fetch the domain category for a given event id,which would be used in raising accounting in case of non-domain category like CH/AT. @param eventId @return @throws FatalException	This method would call the product service to fetch the domain category for a given event id,which will be used in raising accounting in case of non-domain category like CH,AT.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveSingleStatementEntryFlag	This method derives the single statement entry flag @param accountId @return	<p>&lt;p&gt; This function will be used when a customer wishes to deposit cash and cheques simultaneously in a single transaction. &lt;/p&gt; &lt;p&gt; Bank can decide whether to display the transaction as a single entry in Account Statement as well as Statement inquiry screen in OBP. Also Bank can decide whether to charge a single fee for multi fund deposit transaction or charge separate fees for each cheque deposit and cash deposit in a Multi fund deposit transaction. Please note cheque deposit transaction is a non-fee event in product space however in consulting space it is a fee event. &lt;/p&gt; &lt;p&gt; TODO: Based on offer level parameter, the fee is charged. The parameter may indicate single fee for the entire transaction If Multi Fund transaction is not restricted for the offer, system will enable the accordion <input type="checkbox"/> Multi Fund Deposit Parameters <input type="checkbox"/>. This accordion will have following 2 check boxes &lt;/p&gt; &lt;p&gt; 1. Charge Single Fee for a Multi Fund Transaction &lt;/p&gt; &lt;p&gt; 2. Display Single Transaction in Bank Statement &lt;/p&gt; &lt;p&gt; &lt;u&gt;Charge Single Fee for a Multi Fund Transaction&lt;/u&gt; &lt;/p&gt; &lt;p&gt; If <input type="checkbox"/> Charge Single fee for a Multi Fund Deposit <input type="checkbox"/> checkbox is checked, system will charge single fee for Multi fund transaction. System will fetch fee for the event Multi Fund Deposit transaction. This fee can be negotiated if defined in price definition. &lt;/p&gt; &lt;p&gt; If this checkbox is not checked, system will charge fee for each cheque deposit and cash deposit in Multi fund transaction. System will fetch fee associated with events Cash deposit and Cheque Deposit. Please note in this case Fee associated with cheque deposit and cash deposit cannot be negotiated even though it may be defined as a Negotiable fee. &lt;/p&gt; &lt;p&gt; &lt;u&gt;Display Single Transaction in Bank Statement&lt;/u&gt; &lt;/p&gt; &lt;p&gt; If <input type="checkbox"/> Display Single</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>Transaction in Bank Statement <input checked="" type="checkbox"/> checkbox is ticked, system will display the transaction as a single entry in Account Statement as well as Statement inquiry screen in OBP. Please note this cumulative single entry will not include fee amount / amounts. Fee entry / entries will be displayed separately in the statement /Statement inquiry screen. &lt;/p&gt; &lt;p&gt; If this check box is not checked, system will display separate entries for each cheque deposit and cash deposit in statement and statement inquiry screen for Multi fund deposit transaction. &lt;/p&gt; The approach taken for the processing is. &lt;p&gt; The payments screen will get the details of transaction to be performed. The details include &lt;/p&gt; &lt;ul&gt; &lt;li&gt;Cheque of the banks other than own bank.&lt;/li&gt; &lt;li&gt;Cheque of the own bank.&lt;/li&gt; &lt;li&gt;Cheque where amount is less than threshold (Direct credit to DDA.)&lt;/li&gt; &lt;li&gt;Manager Cheques of same bank&lt;/li&gt; &lt;li&gt;Manager Cheques of different banks&lt;/li&gt; &lt;/ul&gt; Now there are 2 approaches possible to cater given requirements. &lt;p&gt; 1.No change in any service and payments can call in loop as per existing flows.CASA can accept the statement flags for controlling statement behaviour. &lt;/p&gt; But after each credit &lt;ul&gt; &lt;li&gt;If there are holds on CASA they will get knocked off in the processing.Which will log the statement entries which will cause the problem in running balance.&lt;/li&gt; &lt;li&gt;CASA will raise the accounting after each processing which needs to be stopped as payment needs amount tag info to add their details.&lt;/li&gt; &lt;/ul&gt; So this approach is not taken for the processing. &lt;/p&gt; Now other approach is Payments will identify the different types of cases possible on CASA and fills them in separate arrays. There are 3 arrays possible &lt;ul&gt; &lt;li&gt;ClearingTransactionDTO:=The array of</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>ClearingTransactionDTO {@link ClearingTransactionDTO} indicates the unclear transaction going to happen on CASA account.</p> <ul style="list-style-type: none"> <li>CashTransactionDTO:= The array of CashTransactionDTO {@link CashTransactionDTO} indicates the cash transaction going to happen of CASA account.</li> <li>FundDepositDTO := The array of FundDepositDTO {@link FundDepositDTO} indicates the credit which are going to happen on CASA account</li> </ul> <p>Now for each DTO DDA will call respective services for performing credits</p> <ul style="list-style-type: none"> <li>{@link DemandDepositCashTransactionService#depositCashApplication(CashTransactionDTO, Boolean)}</li> <li>{@link ChequeClearingApplicationService#updateBalanceOnChequeDepositInternal(ClearingTransactionDTO, Boolean)}</li> <li>{@link DDATransactionApplicationService#creditAccountApplication(AccountTransactionDetailsDTO, boolean)}</li> </ul> <p>Each method while calling credits set {@link FinancialEventType} as RegularWithoutHold.The RegularWithoutHold will not include Earmarks in the processing.For the unclear related transactions it will not perform reapportion on account.So Earmarks will not be apportioned.</p> <p>This method logic works as follows</p> <ul style="list-style-type: none"> <li>It validates the mandatory parameters passed by the caller by using method {@code validateMandatoryParameters }</li> <li>Then the call is given to derive the singleStatement entry flag from the product by using the {@code IDDAProductAdapter}</li> <li>Now the obtained value of flag is passed to respective cash ,credit and unclear credits for further processing along with</li> </ul>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>response object.</p> <ul style="list-style-type: none"> <li>Each method is suppose to fill the response with array of respective method response.</li> <li>The respective method response should contain the original request and amount tags corresponds to that request.</li> <li>Then if entry for singleStatementEntry flag is true then it calls generating 1 statement entry from the state of account</li> <li>Then this method process fees for given transaction based on chargeFeeParameter defined at the offer level.</li> <li>If that parameter is true then fee of MFD full transaction is charged otherwise individual fee is charged</li> </ul> <p>link  <a href="#">@linkplain</a>  MultiFundDepositApplication#processFees(MultiFundDepositDTO)}  @param multiFundDepositDTO @throws FatalException</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProducerAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProducerAdapter	deriveSingleStatementEntryFlag	This method derives the single statement entry flag @param accountId @return	<p>In reverse flow Service expects caller to give no of cheques deposits and original reference number. The {@link ClearingTransactionDTO} contains array of {@link ClearingTransactionDTO#getImmediateCredit()} which contains per cheque how many immediate credits have occurred which needs to be reversed as part of MFD reverse. The system works in following logical flow. &lt;ol&gt; &lt;li&gt;System determines single statement entry is required or multiple by using {@link #deriveSingleStatementEntryFlag(String)}&lt;/li&gt; &lt;li&gt;System has to reverse these transactions.&lt;/li&gt; &lt;/ol&gt; &lt;ul&gt; &lt;li&gt;Cheque deposit &lt;li&gt;Immediate Credits happened on account &lt;li&gt;Direct credits given to account for less than threshold amount. &lt;li&gt;Cash deposit &lt;li&gt;Fund transfer &lt;li&gt;Fee for individual transaction &lt;li&gt;Fee for MFD as a whole &lt;/ul&gt; &lt;ol&gt; &lt;li&gt;Point no 1,2 and 3: &lt;ul&gt; &lt;li&gt;The method in signature expects {@link ClearingTransactionDTO} which contains state of cheques along with Immediate credits happened against those cheques&lt;/li&gt; &lt;li&gt;System will call {@link ChequeClearingApplicationService#updateBalanceOnChequeReturnBeforeValueDateWithPseudo(com.ofss.fc.app.context.SessionContext, ClearingTransactionDTO, Boolean)} .&lt;/li&gt; &lt;li&gt;This will take care for reversal of 1.Cheque Deposit 2.Immediate credits given against those cheques. 3.Direct credits given to account against those cheques because cheque is less than threshold.&lt;/li&gt; &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Point no 4 and 5: &lt;ul&gt; &lt;li&gt;Based on transaction reference number it will fetch list of {@link TransactionAuditItem} using {@link DemandDepositFinancialService#validateForReversal}&lt;/li&gt; &lt;li&gt;Now based on Event id present at {@link</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>TransactionAuditItem#getTransactionAttributes()  }.eventId. It derives equivalent service id and calls  {@link  DemandDepositFinancialService#reverseCreditFunds(  com.ofss.fc.app.context.SessionContext,  TransactionAuditItem[],  com.ofss.fc.domain.transaction.entity.TransactionAttri  butes, String) } It sets up isPseudoEntry flag in {@link  TransactionAttributes#isPseudoEntry()} from output  of {@link #deriveSingleStatementEntryFlag(String)}  &lt;/ul&gt; &lt;/li&gt; &lt;li&gt;Point no 6 and 7: &lt;ul&gt; &lt;li&gt;Check  for fees defined at which level Individual level or MFD  as a whole by using {@link  #isFeeAtMFDLevel(String)}&lt;/li&gt; &lt;li&gt;Derive service  id's applicable to transactions based on event id.In  CASA we may have fees of CASH Deposit or Cheque  deposit.CASA will process for fee for MFD and CASH  Deposit.&lt;/li&gt; &lt;li&gt;Prepare {@link  AccountReversalTransactionDetailsDTO} based on  event id.&lt;/li&gt; &lt;li&gt;Call {@link  DemandDepositFinancialService#manageReverseFees(  com.ofss.fc.app.context.SessionContext,  AccountReversalTransactionDetailsDTO, String)} for  reverse of fees.&lt;/li&gt; &lt;/ul&gt; &lt;/ol&gt; @param  referenceNumber @param clearingTransactionDTO  @throws FatalException</p>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	deriveSingleStatementEntryFlag	This method derives the single statement entry flag @param accountId @return	This method derives the singleStatementEntry applicable for given account for given offer and product by using IDDAProductAdapte#deriveSingleStatementEntryFlag . @param accountId

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	deriveSingleStatementEntryFlag	This method derives the single statement entry flag @param accountId @return	This method derives the singleStatementEntry flag for given account id.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	deriveSingleStatementEntryFlag	This method derives the single statement entry flag @param accountId @return	This method will whether fee for MFD transaction is defined.Returns true if fee to be charged is at MFD level.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isFeeAtMFDLevel	This method derives whether the fee is at MFD level or not. @param accountId	This method suggests whether a single fee is applicable for the entire multi-fund deposit transaction or not. @param accountId @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isFeeAtMFDLevel	This method derives whether the fee is at MFD level or not. @param accountId	<p>This method processes fee for all the transaction which happens in the flow.The fee flow first reads the fee configuration which indicates the fee to be charged as single fee of MFD or Fee of all transactions needs to be charged separately. </p> <p> The callers before calling MFD has to perform all this analysis and should decide whether fee charging is to be done separately.The caller has to generate and pass price ticket number in individual DTO i.e. {@link CashTransactionDTO},{@link FundDepositDTO},{@link ClearingTransactionDTO} for charging the fee separately or need to pass it in {@link MultiFundDepositDTO#getWrappedPriceFeatureDTO()}.This Wrapped price feature may be used to pass some additional parameters. </p> @param multiFundDepositDTO

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	isFeeAtMFDLevel	This method derives whether the fee is at MFD level or not. @param accountId	This method will inquire fee defined at MFD level or not. @param accountId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchEventDescriptionFromEvent	This method would call the product service to fetch the event description for a given event id.	This method would actually log the activity in event processing for periodic fee event alert @param chargingDate @param feeAmount @param accountId @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchEventDescriptionFromEvent	This method would call the product service to fetch the event description for a given event id.	This method would call the product service to fetch the event description for a given event id.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAOfferName	This method returns the name of the offer for given offer id. @param offerId @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAOfferName	This method returns the name of the offer for given offer id. @param offerId @return	This method converts prepare AccountFeederWrapperDTO with mandatory values. @param accountId @return AccountFeederWrapperDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchCASAOfferName	This method returns the name of the offer for given offer id. @param offerId @return	This method will actually invoke the collection module with the new offer details @param demandDepositAccount @param accountBalanceRateDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchBundleName	This method will fetch the bundle name for the given bundle code,it will internally call the OfferBundleApplicationService to fetch the bundle name.	This method fetch bundle name from the bundle service via adapter. @param sessionContext @param preferencesDTO @param demandDepositAccount

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchEligibleOffersForProductCode	This service will fetch a list of eligible offers for the product group. The eligibility check is against a list of party facts for existing party and facility facts in case of lending product groups. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI,OfferInquiryDTO,offerInquiryDTO, , Contains product group, party and facility details. @return OfferInquiryResponse, Returns a list of offers that are applicable for a product group. @throws FatalException	<p>This method makes a call to the restriction service to validate the eligibility rules for the offer swap action. </p> @param ddOfferSwapBusinessPolicyDTO
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This validation will do following:    Check the minor account status of the account    The issuance of chequebook to a party depends on the Minor Account Status @ link com.ofss.fc.enumeration.dda.MinorAccountStatus}     There are three possible statuses at account level.  1) Major  2) MinorWithoutRestrictions  3) MinorWithRestrictions    1)Major: It will allow to perform the cheque book request.    2)MinorWithoutRestrictions:   There are two parameters at the bank policy level.  a) relation required  b) relation type    When relation Flag = FALSE  If the relation required flag is false then there is no restriction for this status and cheque book request will be honored.    When relation Flag = TRUE  Validate if the

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
				<p>relation type defined at the bank policy level is present in the party account relationship for this account.&lt;br /&gt; &lt;br /&gt; If present then allow cheque book request.&lt;br /&gt; &lt;br /&gt; If account DOES NOT have relationship present then &lt;br /&gt; throw business validation excetion.&lt;br /&gt; &lt;br /&gt; For example relation required flag is TRUE and relationship defined is Guardian then this account should have a guardian relation ship maintained for the account.&lt;br /&gt; &lt;br /&gt; 3)MinorWithRestrictions:&lt;br /&gt; There are two separate paramters maintained at bank policy level for this status as well. The rest of the processing is exactly same as MinorWithoutRestrictions.&lt;br /&gt; &lt;br /&gt; The derivation of the Account Minor status is created at account opening itself and it is updated if a parties minor status changes in batch.&lt;br /&gt; &lt;br /&gt; &lt;br /&gt;Validation for auto reorder of chequebook.&lt;br /&gt; The same rules of chequebook request for minor status will also be applicable for reorder of chequebook.&lt;br /&gt; @param account</p>
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method will check if the party is minor and the operating instruction channel access type is no for given channel then return true,false otherwise @param sessionContext @param partyAccountRelationshipDetailsDTO @param operatingInstructionChannel @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	This method will check if the party is operationally major and the operating instruction channel access type is no for given channel then return true,false otherwise @param partyAccountRelationshipDetailsDTO @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchPartyBankPolicy	<p>This method will fetch the party bank policy definition for the input bank code. </p> @param bankCode @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchFeeRecognitionBranchTypeFromEvent	This method is used to fetch from Event Table the Fee Recognition Branch Type for a given EventID	This method calls the adapter to fetch the Fee Recognition Branch Type for a given Event Id from PM Events Table @param eventId @return FeeRecognitionBranchType @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchFeeRecognitionBranchTypeFromEvent	This method is used to fetch from Event Table the Fee Recognition Branch Type for a given EventID	This method is used to store fee recognition branch details. <ol> <li> Fee is income for the branch.In accounting, fee can be recognised by account branch or transaction branch. <li> For Fee type of arrear, we store fee recognition branch details. <li> Fee recognition branch type is stored per event level in PMU table. </ol> @param arrearInformation @param arrear
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchFeeRecognitionBranchTypeFromEvent	This method is used to fetch from Event Table the Fee Recognition Branch Type for a given EventID	This method will fetch the fee recognition branch type based on the event id passed to it from the PMU Events table
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	This method derives the pickupDate of a recurring instruction. This method works as follows :  <pre> Checks from commonServices whether the nextExecutionDate is a holiday or not. If it is not a holiday then return pickupDate = nextExecutionDate If it is a holiday then : It picks the holiday treatment defined at the bank level. If Previous fetch the last working day which falls prior of nextExecutionDate If Next fetch the next working day of the bank which falls after nextExecutionDate. </pre> @param nextExecutionDate @param fundsFlowType @param holidayTreatmentMethod @return

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	This method will be used to update all the standing instruction's pickup date when calendar day has been marked as holiday from working. @param affectedDate @param branchId
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	Validates the max number of casa accounts that are allowed to be offset for a CASA beneficiary.
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	This method is used to create new rate variance for the cloned account. The UserDiscretionaryMargin for fixed and variable is copied from the parent account id to the cloned account Id these rates are then merged to RateVarianceToRecommendation while creating the pricing Plan for the cloned account,this method will also check if campaign udm's are true in bank policy and the variance type is of campaign type,if yes then in case of cloning campaign udm's will be carried forward,else it will not be carried forward.If the variance type is not campaign type then carry forward the UDM's for the account which is getting cloned. @param wrappedIndexFeatureDTOs @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	This method will check whether in case of offer swap,there is any campaign udm defined in EventRateVarianceDTO and the bankpolicy definition allows to carry forward campaigning udm's,if yes the campaign UDM's will be carried forward in case offer swap happens else campaign UDM's will be skipped while intimating pricing.In case the variance type is not campaign udm then carry forward the UDM's. @param eventRateVarianceDTOs @param demandDepositAccount @return @throws FatalException



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	Fetch applicable UDM to carry forward in amend OD. @param pricingTicketNumber @param eventRateVarianceDTOs @param demandDepositAccount @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	The method will call Fee management service to charge fee for the OD limit increase/decrease event. @param demandDepositAccount @param currentSanctionedLimitAmount @param newSanctionedLimitAmount @param limitAmount
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	This method decides whether to create new recommendation or not. Following are the steps involved. <ol> <li>Increase Limit Amount: Reprice if account is in fixed Rate period parameter is checked / Variable rate period parameter is checked <li>Decrease Limit Amount: Reprice if account is in fixed Rate period parameter is checked / Variable rate period parameter is checked. <li>Offer Swap Increase Limit Amount / Decrease Limit Amount +Offer swap : In this case irrespective of reprice parameter being yes or no, system will reprice the account for which offer swap is to be done. <li>Increase / Decrease Limit Term / Link new collateral to facility /De link existing collateral /add or remove borrower - In these case reprice parameter is not applicable. </ol> @param pricingRequestDTO @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	True value will indicate that the net amount will exclude the temporary excess limit amount from the available balance of the account. False value will indicate that add the temporary excess limit amount in calculating the net amount.

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchDDABankPolicy	<p>This method will fetch the dda bank policy definition for the input bank code. </p> @param bankCode @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	This method would validate the allowed backdated transaction days has not breached as per the bank policy definition. @param bankCode @param valueDate @param processDate
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchLoanBankPolicy	<p>This method will fetch the loan bank policy definition for the input bank code. </p> @param bankCode @return	
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchRepriceDetailsForEvent	This method reprice event details for particular event in demand deposit module. @param applicationContext @param bankCode @param eventCode @return	This method takes decision about repricing on domain level fact change. <ol> <li>Bank level repricing policy is maintained for each event. <li>Reprice parameters are maintained for fixed period and variable period. <li>Account can be in fixed or variable period. Respective parameters are considered. </ol> @param demandDepositAccount @param eventId @param onDate @return
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	checkIfOffersValid	This service check if the offer passed is valid and not expired. @fcb.param MI,SessionContext,sessionContext, ,Session Context which must contain the session context information such as user id, branch, branch code, channel etc. @fcb.param MI, String offerId, , Indicates the offer identifier being validated @return Boolean, if offer is valid and not expired. @throws FatalException	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchIssuedByBankFlag	This flag indicates that the whether bank can issue cheque or not. If true then bank issued cheques are only honoured. If false then any cheque number even if not issued by bank is supported. @param bankCode @return	This method will make a call to the Casa Bank Policy Parameters Application Service with input as bank Code to verify that bank supports Customer Issue Check or not. If the return value false indicates only bank can issue the cheques and customer issue cheques are not allowed. else Bank will not issue cheque book, customer can present their own issued cheques. @param sessionContext @return @throws FatalException
IDDAProductAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\dda\IDDAProductAdapter	fetchIssuedByBankFlag	This flag indicates that the whether bank can issue cheque or not. If true then bank issued cheques are only honoured. If false then any cheque number even if not issued by bank is supported. @param bankCode @return	This method will populate the bank policy definition by calling Casa bank policy parameters Application Service. @param sessionContext @return casaBankPolicyParametersInquiryResponse @throws FatalException
IPricingTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAAdapter	repriceTDDeposit	This method will be used to reprice the TD account whenever there is a change in Fee PPC @param sessionContext @param arrangementNumber @param repriceDate	
IPricingTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAAdapter	repriceTDDeposit	This method will be used to reprice the TD account whenever there is a change in Fee PPC @param sessionContext @param arrangementNumber @param repriceDate	This method will reprice given TD Account @param sessionContext @param arrangementNumber @param repriceDate @throws FatalException
IPricingTDAAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAAdapter	fetchTDAccountsForProductOffer	This method will be used to reprice the TD account whenever there is a change in Fee PPC @param sessionContext @param arrangementNumber @param repriceDate	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPricingTDA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAadapter	fetchTDAccountsForProductOffer	This method will be used to reprice the TD account whenever there is a change in Fee PPC @param sessionContext @param arrangementNumber @param repriceDate	This method fetches the list of accounts that are to be repriced due to the addition of new PPC. It calls the respective account module with the list of product Offer Ids. @param sessionContext @param domainCategoryId @param groupList @return list of accounts @throws FatalException
IPricingTDA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAadapter	getAllDepositsByPostingDate	This method will return all Deposits which have been opened on the given value date. @param valueDate @return @throws FatalException	
IPricingTDA dapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\pricing\IPricingTDAadapter	getAllDepositsByPostingDate	This method will return all Deposits which have been opened on the given value date. @param valueDate @return @throws FatalException	This method is used to fetch all accounts opened today @param domainCategoryID, dueDate @return List: accountOpenedToday @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	addUser	<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	addUser	<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	addUser	<p>This represents the addUser method of the entity.&lt;br&gt;This method takes the IdentityDTO as the input. It validates the DTO&lt;br&gt; to make sure all the mandatory fields are present.&lt;br&gt; It creates a new user in the OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	updateUser	<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	updateUser	<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the updateUser method of the entity.&lt;br&gt; It updates the attributes of the user in the OIM.&lt;br&gt; It does not update the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes. @fcb.param MI,IdentityDTO,oracleIdentityDTO, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	changeLoginEmailId	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	changeLoginEmailId	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt;</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the changeUserId method of the entity.&lt;br&gt;This method changes the user Id of the user.&lt;br&gt; The userId is the current user Id and newUserId is the changed user id.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @fcb.param MI,String,newUserId, ,It represents the new User id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	changeLoginEmailId	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	changeLoginEmailId	<p>This represents the changeLoginEmailId method of the entity.&lt;br&gt; It changes the email id of the user in OIM.&lt;br&gt;</p> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,loginEmailId, ,It represents the login Email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the changeUserId method of the entity.&lt;br&gt;This method changes the user Id of the user.&lt;br&gt; The userId is the current user Id and newUserId is the changed user id.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @fcb.param MI,String,newUserId, ,It represents the new User id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	changeUserId	<p>This represents the changeUserId method of the entity.&lt;br&gt;This method changes the user Id of the user.&lt;br&gt; The userId is the current user Id and newUserId is the changed user id.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @fcb.param MI,String,newUserId, ,It represents the new User id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the changeUserId method of the entity.&lt;br&gt;This method changes the user Id of the user.&lt;br&gt; The userId is the current user Id and newUserId is the changed user id.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @fcb.param MI,String,newUserId, ,It represents the new User id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUserWithCriteria method of the entity. This method fetches the details of the user based on some criteria.  Following two criteria are allowed for the user inquiry :  <ul> <li>1. Users.User ID</li> <li>2. Users.Email</li> </ul> @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes. @fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.  @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	<p>This represents the inquireUser method of the entity.&lt;br&gt;This method fetches the details of the given user from the OIM.&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt;@return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUserWithUserID method of the entity.&lt;br&gt;This method fetched the details of the user based on user's email Id.&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,emailId, ,It represents the email Id of the user.&lt;br&gt;@return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	inquireUser	<p>This represents the inquireUser method of the entity.&lt;br&gt;This method fetches the details of the given user from the OIM.&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user. @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user. @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	This represents the inquireUser method of the entity. This method fetches the details of the given user from the OIM. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,userId, ,It represents the current User Id of the user. @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id. @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system. The information is used by the system for authentication purposes. @fcb.param MI,String,emailId, ,It represents the email Id of the user. @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	<p>This represents the inquireUser method of the entity.&lt;br&gt;This method fetches the details of the given user from the OIM.&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :&lt;br&gt; &lt;ul&gt; &lt;li&gt;1. User Login&lt;/li&gt; &lt;li&gt;2. Email&lt;/li&gt; &lt;li&gt;3. USERS_PARTY_ID&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUser	<p>This represents the inquireUser method of the entity.&lt;br&gt;This method fetches the details of the given user from the OIM.&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, ,It represents the current User Id of the user.&lt;br&gt;@return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt;The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUserWithEmailID	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUserWithEmailID	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the inquireUserWithUserID method of the entity. This method fetched the details of the user based on user's email Id.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,emailId, ,It represents the email Id of the user.  @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIIdentityAdapter	inquireUserWithCriteria	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :&lt;br&gt; &lt;ul&gt; &lt;li&gt;1. Users.UserID&lt;/li&gt; &lt;li&gt;2. Users.Email&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :&lt;br&gt; &lt;ul&gt; &lt;li&gt;1. Users.UserID&lt;/li&gt; &lt;li&gt;2. Users.Email&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	inquireUserWithCriteria	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :</p> <ul style="list-style-type: none"> <li>1. Users.UserID&lt;/li&gt;</li> <li>2. Users.Email&lt;/li&gt;</li> </ul> <p>@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUserWithUserID method of the entity.&lt;br&gt;This method fetched the details of the user based on user's email Id.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,emailId, ,It represents the email Id of the user.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUserWithCriteria	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :&lt;br&gt; &lt;ul&gt; &lt;li&gt;1. Users.UserID&lt;/li&gt; &lt;li&gt;2. Users.Email&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUserWithCriteria method of the entity.&lt;br&gt;This method fetches the details of the user based on some criteria.&lt;br&gt; Following two criteria are allowed for the user inquiry :&lt;br&gt; &lt;ul&gt; &lt;li&gt;1. User Login&lt;/li&gt; &lt;li&gt;2. Email&lt;/li&gt; &lt;li&gt;3. USERS_PARTY_ID&lt;/li&gt; &lt;/ul&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,criteria, ,It represents the search criteria for inquiring user.&lt;br&gt; @fcb.param MI,String,criteriaValue, ,It represents the value of the search criteria for inquiring users.&lt;br&gt; @return {@link com.ofss.fc.app.sms.service.identity.IdentityInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUsers	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;  @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;  @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	inquireUsers	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;  @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the inquireUsers method of the entity.&lt;br&gt;This method fetches information about all the Users that were created on that particular Day&lt;br&gt;  @fcb.param  MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.  @fcb.param MI,String,date, ,Date for which the addition of users list is to be inquired yyyyMMDD format  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	resetPassword	This represents the resetPassword method of the entity. This method resets the password of the user.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @fcb.param MI,String,password, ,It represents the password of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the resetPassword method of the entity. This method resets the password of the user.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	resetPassword	<p>This represents the resetPassword method of the entity.&lt;br&gt;This method resets the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,password, ,It represents the password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the resetPassword method of the entity.&lt;br&gt;This method resets the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	setPassword	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the resetPassword method of the entity.&lt;br&gt;This method resets the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	setPassword	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	setPassword	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt; @fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This method validates the user password.&lt;br&gt; If the password is null or an empty string, it sets the default password.&lt;br&gt; @param identityDTO</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	setPassword	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the resetPassword method of the entity.&lt;br&gt;This method resets the password of the user.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	setPassword	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the setPassword method of the entity.&lt;br&gt;This method sets the password of the user.&lt;br&gt; Here the currentPassword is the current password of the user and&lt;br&gt; and the newPassword is the changed password.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @fcb.param MI,String,currentPassword, ,It represents the current password of the user.&lt;br&gt; @fcb.param MI,String,newPassword, ,It represents the new password of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	lockAccount	<p>This represents the lockAccount method of the entity.&lt;br&gt;This method locks the account of the user in OIM.&lt;br&gt; If the account of the user is locked, then the user won't be able to login to his account&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the lockAccount method of the entity.&lt;br&gt;This method locks the account of the user in OIM.&lt;br&gt; If the account of the user is locked, then the user won't be able to login to his account&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, ,It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIIdentityAdapter	lockAccount	<p>This represents the lockAccount method of the entity.&lt;br&gt;This method locks the account of the user in OIM.&lt;br&gt; If the account of the user is locked, then the user won't be able to login to his account&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the lockAccount method of the entity.&lt;br&gt;This method locks the account of the user in OIM.&lt;br&gt; If the account of the user is locked, then the user won't be able to login to his account&lt;br&gt;@fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.&lt;br&gt;@fcb.param MI,String,userId, It represents the User Id of the user.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	unLockAccount	This represents the unLockAccount method of the entity. This method unlocks the account of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the unLockAccount method of the entity. This method unlocks the account of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	unLockAccount	This represents the unLockAccount method of the entity. This method unlocks the account of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the unLockAccount method of the entity. This method unlocks the account of the user in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	maintainUsers	<p>This represents the maintainUsers method of the entity.&lt;br&gt;This method maintain users from OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,userManagementDTOs array ,userManagementDTOs, ,It represents the usersDTOs for management.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	<p>This represents the deletes bulk users method of the entity.&lt;br&gt; It deletes multiple users in OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,IdentityDTOs,identityDTOs, ,It represents the DTO corresponding to IdentityDTO.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\sms\IIdentityAdapter	maintainUsers	<p>This represents the maintainUsers method of the entity.&lt;br&gt;This method maintain users from OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,userManagementDTOs array ,userManagementDTOs, ,It represents the usersDTOs for management.&lt;br&gt; @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for &lt;i&gt;Transaction Failure&lt;/i&gt; section</p>	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIdentityAdapter	fetchRoles	This represents the fetchRoles method of the entity. This method maintain users from OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,userId ,It represents the userId.  @return {@link UserInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the deletes bulk users method of the entity.  It deletes multiple users in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,String,userId, ,It represents the search filter of the User Id of the user.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIdentityAdapter	fetchRoles	This represents the fetchRoles method of the entity. This method maintain users from OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,userId ,It represents the userId.  @return {@link UserInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<i>Transaction Failure</i> section	
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIdentityAdapter	fetchRoles	<p>This represents the fetchRoles method of the entity.&lt;br&gt;This method maintain users from OIM.&lt;br&gt; @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.&lt;br&gt; The information is used by the system for authentication purposes.</p> <p>@fcb.param MI,userId ,It represents the userId.&lt;br&gt; @return {@link UserInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for</p>	<i>Transaction Failure</i> section

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIdentityAdapter	searchUsers	This represents the searchUsers method of the entity. This method maintain users from OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,userId ,It represents the userId.  @return {@link UserInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section	This represents the deletes search users method of the entity.  It searches multiple users in OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,IdentityDTOs,identityDTOs, ,It represents the DTO corresponding to IdentityDTO.  @return {@link com.ofss.fc.service.response.TransactionStatus} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for <i>Transaction Failure</i> section
IIdentityAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\sms\IIdentityAdapter	searchUsers	This represents the searchUsers method of the entity. This method maintain users from OIM.  @fcb.param MI,com.ofss.fc.app.context.SessionContext,sessionContext, , This parameter captures the basic information about the user of the system.  The information is used by the system for authentication purposes.  @fcb.param MI,userId ,It represents the userId.  @return {@link UserInquiryResponse} @throws com.ofss.fc.infra.exception.FatalException Please refer to the documentation above for	

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
			<i>Transaction Failure</i> section	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipUpdate	Update the PartyAccountRelationship for a party in Collections module. @param	This method adds {@code PartyToAccountRelationshipDTO} in the collections module whenever it gets added in the party table. @param sessionContext @param accountRelationshipDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipUpdate	Update the PartyAccountRelationship for a party in Collections module. @param	This method deletes {@code PartyToAccountRelationshipDTO} from the collections module whenever it gets deleted from the party table. @param sessionContext @param accountRelationshipDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipUpdate	Update the PartyAccountRelationship for a party in Collections module. @param	This method updates the {@code PartyToAccountRelationshipDTO} in the collection module. @param sessionContext @param partyAccountRelationshipDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipUpdate	Update the PartyAccountRelationship for a party in Collections module. @param	This method updates the array {@code PartyToAccountRelationshipDTO} in the collection module. @param sessionContext @param partyAccountRelationshipDTOs @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyDemographicsUpdate	Update the PartyDemographics for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAddressUpdate	Update the PartyAddress for a party in Collections module. @param	This method updates the address in collection module whenever the address gets updated in OBP @param sessionContext @param partyAddressDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyEmploymentUpdate	Update the PartyEmploymentHistory details for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyContactPointUpdates	Update the PartyContactPoints for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyIdentificationUpdates	Update the PartyIdentifications for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyWarningIndicatorUpdates	Update the PartyWarningIndicator for a party in Collections module. @param sessionContext @param partyWarningIndicatorDTO @param memoList @param modifiedPartyWarningIndicatorDTO @throws FatalException	This method publishes warning indicator changes to collection. @param sessionContext @param partyWarningIndicatorDTO @param memoList @param modifiedPartyWarningIndicatorDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyNameUpdates	Update the PartyNames for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAddressAdd	Add the PartyAddress for a party in Collections module. @param	This method adds the address in collection module whenever the address gets adds in OBP @param sessionContext @param partyAddressDTO @throws FatalException

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAddressesAdd	Add the PartyAddress for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyEmploymentAdd	Add the PartyEmploymentHistory details for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyEmploymentAdd	Add the PartyEmploymentHistory details for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyContactPointAdd	Add the PartyContactPoints for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyContactPointAdd	Add the PartyContactPoints for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyIdentificationAdd	Add the PartyIdentifications for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyIdentificationAdd	Add the PartyIdentifications for a party in Collections module. @param	



Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyNameAdd	Add the PartyNames for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyNameAdd	Add the PartyNames for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	addPartyImage	Method to feed new photo or signature image linked to the party to collections @param sessionContext @param imageDTO @throws FatalException	Feed image to collections @param sessionContext @param imageDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	updatePartyImage	Method to feed updated photo or signature image linked to the party to collections @param sessionContext @param imageDTO @throws FatalException	Feed image to collections @param sessionContext @param imageDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAddressForDelete	Delete the future PartyAddress for a party in Collections module. @param	This method deleted the address in collection module whenever the address gets deleted in OBP @param sessionContext @param partyAddressDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAddressForDelete	Delete the future PartyAddress for a party in Collections module. @param	
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyIdentificationDelete		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyIdentificationDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyContactPointDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyContactPointDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyEmploymentDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyEmploymentDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipDelete		This method deletes {@code PartyToAccountRelationshipDTO} from the collections module whenever it gets deleted from the party table. @param sessionContext @param accountRelationshipDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipDelete		

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipAdd		This method adds {@code PartyToAccountRelationshipDTO} in the collections module whenever it gets added in the party table. @param sessionContext @param accountRelationshipDTO @throws FatalException
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyAccountRelationshipAdd		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyNameDelete		
IPartyCollectionAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\party\IPartyCollectionAdapter	partyNameDelete		
IDMSWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IDMSWebServicesAdapter	processWithResponse		
IDMSWebServicesAdapter	com.ofss.fc.app.adapter.internal.interface.jar@\com\ofss\fc\app\adapter\ops\IDMSWebServicesAdapter	processWithResponse		This method enables the MTOM client support and invokes the doCallIDS method of the Documaker CompositionService. The sample code to enable the MTOM at the client end is as below: SOAPBinding binding = (SOAPBinding)((BindingProvider) compositionServicePortType).getBinding(); binding.setMTOMEnabled(true);

Adapter Name	Adapter Path	Adapter Method	Interface Method Javadoc	Implementation Method Javadoc
IDocumentContactPointPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentContactPointPartyAdapter	setContactPoint	Sets the contact point. @param sessionContext the session context @param documentDeliveryPreferenceDTO the document delivery preference dto @throws FatalException the fatal exception	Gets the delivery details for party.for each passed recipients delivery details are fetched from party. @param sessionContext @param documentDetailDTO @param documentPolicyDetailDTO @return @throws FatalException
IDocumentContactPointPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentContactPointPartyAdapter	setContactPoint	Sets the contact point. @param sessionContext the session context @param documentDeliveryPreferenceDTO the document delivery preference dto @throws FatalException the fatal exception	TODO Gets the delivery details for associate parties. @param sessionContext the session context @param documentDetailDTO the document detail dto @return the delivery details for associate parties
IDocumentContactPointPartyAdapter	com.ofss.fc.app.adapter.internal.interface.jar@com\ofss\fc\app\adapter\content\IDocumentContactPointPartyAdapter	setContactPoint	Sets the contact point. @param sessionContext the session context @param documentDeliveryPreferenceDTO the document delivery preference dto @throws FatalException the fatal exception	This method call the party module to get party level preferences for document delivery.This method will take the relation maintain after execution of the policy in documentTypeRelationshipDTO.this DTO actually contains the name of the relationship such as Applicant/co-applicant/Guaranter. Then for each relationship define for policy level and inside that all the relationship passed by the caller check which are equal then for found relationships find out the parties pssed by the caller.For each party fetch the address details from PartyAddressApplicationService and for Email and and fax call contact application service. @param sessionContext the session context @param documentDetailDTO the document detail dto @param documentPolicyDetailDTO the document policy detail dto @return DocumentDeliveryPreferencesDTO @throws com.ofss.fc.infra.exception.FatalException @throws FatalException the fatal exception